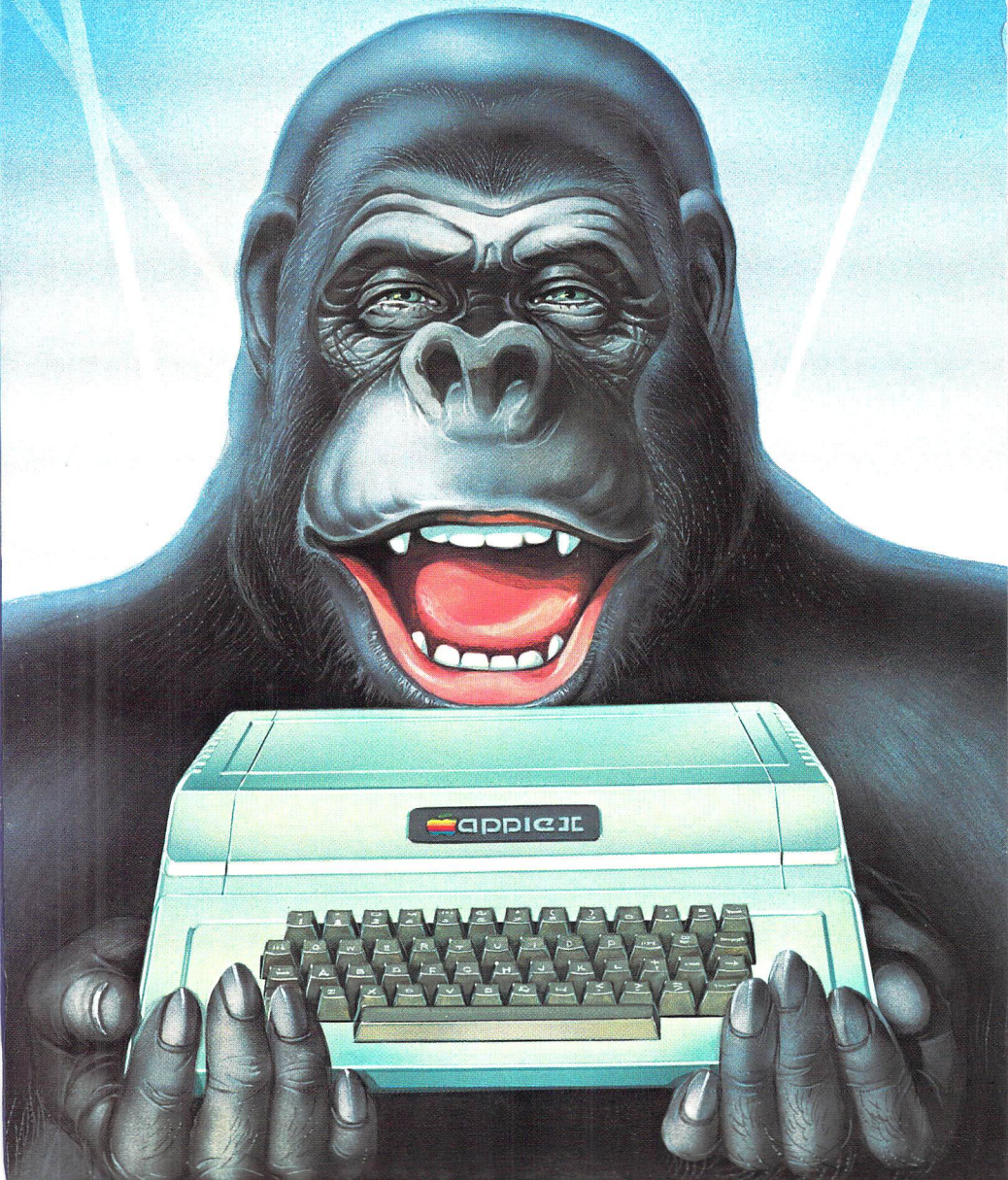
SYBEX

APPLE II®

BASIC PROGRAMS IN MINUTES

STANLEY R. TROST



APPLE II

BASIC PROGRAMS IN MINUTES

APPLE II®
BASIC PROGRAMS IN MINUTES

STANLEY R. TROST

BERKELEY • PARIS • DÜSSELDORF

Cover art by Sato Yamamoto
Design by Ingrid Owen

Library of Congress Card Number: 83-61385
ISBN 0-89588-121-7
Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ACKNOWLEDGEMENTS

Many people contributed ideas and hard work to this book. Rudolph Langer suggested the book and pointed the way to a number of improvements. David Lichtblau provided valuable assistance with program development and verification. Barbara Gordon and Joel Kreisman edited the manuscript, checked the programs, and prepared the programs for production. Many other members of the Sybex staff were also part of the team that put this book together.

Stan Trost
March 1983

CONTENTS

APPENDICES

PREFACE

Apple II BASIC Programs in Minutes will help you discover the full power and flexibility of your Apple computer. This book contains more than 65 practical programs and subroutines that are fully tested and ready to run; they provide a wide variety of business, personal, and educational applications. You do not need to understand the BASIC programming language to use these programs; you just enter them into your computer and use them for your own applications.

These programs will help you make important financial decisions; for example, you can find the break-even point of a new business, calculate the monthly payment necessary to establish a college fund for your children, and find the future value of regular deposits in an individual retirement account (IRA). The record keeping programs allow you to create a mailing list, phone directory, and automobile log. Students and nonstudents alike will find the mathematics practice programs especially useful. These practice programs help sharpen arithmetic skills; they can also help students with homework assignments.

Apple II BASIC Programs in Minutes contains a number of unique programs. The fraction practice programs, long multiplication drill, record keeping programs, and combined data analysis and graphing programs are seldom found in a collection of programs.

The use of standard subroutines is also unique. A primary goal of this book is to provide programs that are *short* and easy to enter in your computer. To achieve this goal, the programs were created using a standard set of programs as subroutines. These subroutines will also save you a great deal of effort when you create your own programs.

This book does not teach BASIC language programming; however, use of the programs will help you understand BASIC in a painless, almost automatic way. As you become familiar with BASIC, you will be able to modify the programs in this book and create new programs.

Let's now review the contents of the book. Chapter 1 gives instructions for using the programs and reviews the interactive features built into them. You should read this chapter before you use the programs.

Chapter 2 is a set of programs for home and business finance. It includes programs that calculate the future value of a bank deposit, the future value of a series of payments (for example, the future value of deposits in an IRA), the interest rate earned on an investment, and the time required to double your money. These programs can help you make many important investment decisions.

Chapter 3 contains a collection of business programs. These programs compute depreciation, the break-even point for a new business, wages earned with overtime, and decision analysis. The business programs will be helpful to business people, employees, and consumers.

In Chapter 4 we present real estate application programs including mortgage payment calculation, balloon payment calculation, and rental property analysis. You will, for example, be able to find the interest you can save by increasing your monthly house payment. Investors, homeowners, and prospective homeowners will find these programs helpful.

Chapter 5 includes a set of data analysis programs that can be used to analyze both scientific and financial data. The programs can compute mean and standard deviation, various moving averages, and linear regression, and they can graph the results. These programs can be of use in analyzing and projecting stock market trends, among other things.

Chapter 6 contains a set of record keeping programs to keep personal and automobile records. With these programs you will be able to use your computer to keep mailing lists and phone directories as well as mileage records for your car. The programs can be modified easily for other applications.

Chapter 7 presents a number of useful arithmetic drills for students of all grade levels and abilities. These drills include addition, subtraction, multiplication, and division, using whole numbers as well as fractions. These programs can help sharpen your math skills and even help with homework!

We also include three appendices with this book. Appendix A lists the standard subroutines that enable the programs within the chapters to be so short. Their functions are described in REM statements. These subroutines are in numerical order so that you can easily refer to them. Appendix B shows you how to use these subroutines to create your own programs. Descriptions and examples are provided.

Appendix C lists several routines that allow you to use the full capabilities of an Epson printer. You can use these programs to gain access to all of the different printing modes available (condensed, expanded, double-strike, and so on).

1 INTRODUCTION

This chapter shows you how *Apple II BASIC Programs in Minutes* can greatly increase your computer's effectiveness. You will learn how to enter and run programs, create a subroutine library, and build a library of related programs that can run from a menu. You will also learn what interactive program design means and how it is accomplished in these programs.

The programs in this book do not require knowledge of the BASIC programming language. They can simply be used as is to solve a wide variety of business, home, and education problems. However, if you wish to expand your programming knowledge, these programs will show you how to take full advantage of the powerful features of your computer.

HOW TO START

You will need an Apple II or Apple IIe computer with 64K of memory and a disk drive. The programs in this book are written in Applesoft BASIC. Review your Applesoft manual for detailed instructions on loading and running BASIC.

Place a DOS diskette in your disk drive and turn on the power. After a short pause the screen displays the cursor:

```
]
```

You should now remove the DOS diskette from the disk drive and put it away.

You will need two formatted disks (formatting instructions are in the Apple DOS manual) to save the programs you enter (one as a master disk and one as a backup). You should always make two copies of every program.

At this point you can enter any of the programs in this book. You enter a program by typing the lines in the program listings exactly as they appear in this book. After each line press the return key. Let's enter the following program:

```
100 PRINT "HELLO"
110 END
```

Begin by typing the first line (100 PRINT "HELLO"). Enter this line by striking the return key. Complete the program entry by typing and entering the second line.

You will sometimes make mistakes in entering the programs. Short lines are easily corrected by retyping the entire line. The Applesoft manual explains other methods you can use to correct errors on longer lines.

To run this program, type RUN 100 and strike the return key. Notice the display on your screen; your program has printed HELLO below the program you entered. We'll give you more information on entering, saving, and running the programs later in this chapter.

REMARK STATEMENTS

We use *remarks* to make the programs easier to understand. Remark statements begin with the symbol REM. In Figure 1.1, remarks make a border at the top of the program and explain the

```
100  REM --------------
101 N$="SAMPLE PROGRAM"
102  REM --------------
110  INPUT X
120  INPUT Y
130  PRINT X+Y: REM PRINT THE SUM OF X AND Y
140  END
```

Figure 1.1: Program Listing Including Remarks

```
100  REM
101 N$="SAMPLE PROGRAM"
110  INPUT X
120  INPUT Y
130  PRINT X+Y
140  END
```

Figure 1.2: Program Listing with Remarks Condensed

action of line 130. With one exception, you don't have to type the remarks as you enter the programs. The exception is that you must type the word REM on the first line of the program. The first REM is necessary because of the design of our programs. Figure 1.2 shows the same program listing after you type the necessary lines.

BUILDING A SUBROUTINE LIBRARY

The programs in this book use a central subroutine library to simplify program entry and output formatting. A *subroutine* is simply a program that is used as part of another program. For example, one central subroutine handles data entry for many of our programs. These subroutines allow our programs to be short and therefore easy for you to enter.

Appendix A contains the program listings for the subroutines you will use in this book. Appendix B shows you how to use some of these subroutines to create your own programs.

The following list shows the subroutines you must enter for the different chapters in this book. You should enter and save the subroutines on disk as you need them; thus you need enter each subroutine only once. The numbers in the list refer to the first line number of each subroutine.

Chapters 2–7	4500, 6000, 6500, 7000, 7500, 8000 8500, 9000, 9500
Chapter 6	4000, 5000, 5500

We will begin by building the subroutine library for Chapters 2 through 5. Turn to Appendix A, where you will find the listings for the standard subroutines. Because the subroutine at line 4000 isn't needed until Chapter 6, begin with line 4500. First type NEW to clear any old programs from the computer; then type the instructions for this subroutine into your computer. (Remember that remarks are not entered except in the first line.) Your entry should match that shown in Figure 1.3.

Continue to enter the rest of the subroutines necessary for Chapters 2–5 and 7. After you enter the standard routines, save them on disk. Follow the instructions in the DOS manual to ready your disk. Then type

```
SAVE SUBLIB
```

Your central subroutine library is now stored with the file name SUBLIB. You will merge SUBLIB with your application programs when you run the applications. You should add the remainder of the subroutines to SUBLIB when you enter the programs in Chapter 6.

```
4500  REM
4660 Q1$="WOULD YOU LIKE TO "
4670 Q3$="AGAIN? (Y OR N)"
4690  GOSUB 6000
4710  GOSUB 9500
4720  RETURN
```

Figure 1.3: Program Listing: ANOTH Subroutine

ENTERING AND RUNNING THE PROGRAMS

You are now ready to enter a program. Type the program instructions as they are shown in the program listings. The programs in Chapters 2, 3, and 4 should be saved in one file for each chapter. (Use file names that are meaningful to you.) The programs can then be run by loading the subroutine library, merging the file for the desired chapter, and typing RUN and the first line number of the desired program. When you have typed in all the programs in a chapter, you can create a menu to run the programs in that chapter. A menu is a list of programs that you can use to select the program you want to run. Instructions for creating the menu are given in each chapter. The programs within each chapter begin with unique line numbers; this allows you to store the programs and menu for each chapter together in one file.

For example, Chapter 2 contains a number of programs that are useful when you make financial calculations. Type in the Future Value of a Deposit program and save it on disk or tape by typing

```
SAVE FIN
```

When you need another financial program, load the first financial program you saved by typing

```
LOAD FIN
```

Now you can type in the next financial program you need. Save both programs together by typing

```
SAVE FIN
```

You can see how this method enables you to build a library of financial programs.

You merge the subroutines and application programs by using the Applesoft program RENUMBER. Place the DOS disk in the disk drive and type RUN RENUMBER. Now put your program disk in the drive and follow the sequence below (be sure to hit the RETURN key after each line):

```
LOAD SUBLIB
&H
LOAD FIN
&M
RUN 300
```

Note: the RENUMBER program takes about two minutes to merge the subroutines and programs. Wait for the cursor before you type RUN.

When you have typed in all the programs in Chapter 2, enter and save the Financial Programs Menu. This will enable you to run any program in that chapter by selecting it from the menu—no more line numbers to remember. To activate the menu program you should replace the END command in the last line of each program with a RETURN command.

The programs in Chapters 5–7 are entered in a slightly different way. They begin rather than end with a menu program, so in these chapters you first enter the menu and then the application program you choose. The programs are run by selecting them from the menu. Because we place the menus at the beginning of each set of programs in these chapters, you don't have to bother with replacing any END statements in the programs.

INTERACTIVE FEATURES

The word *interactive* refers to the interaction between the computer, the programs, and the user. The goal of an interactive program is to make it easy for the user to interact with the computer.

We accomplish this goal in various ways. When possible, we answer questions with a single keystroke that doesn't require the enter key. For example, programs are selected from the menus with a single digit, and yes/no questions require simply a Y or an N. Another interactive feature we use is to place all questions at the bottom of the screen, so you can always look for them in the same place. Your answers are listed at the top of the screen, so you can keep track of the data you have input. We also use sound to alert you to possible errors in input, particularly to incorrect answers in the arithmetic drills.

SUMMARY

This chapter provides you with detailed instructions for creating and saving a subroutine library and for using the library with the rest of the programs in this book. This chapter also tells you how to

create a library of related programs that can be run from a menu. Each time you need a new program, you add it to the programs you have already saved.

The use of the central subroutine library allows the programs in this book to be very short and easy to enter, and the interactive features built into the programs make them easy to use.

2 HOME AND BUSINESS FINANCE

This chapter contains ten programs useful for many home and business finance applications. For example, let's assume that you are going to deposit money in a bank account for a specific term. One bank pays 8% compounded monthly, and another bank pays 8¼% compounded quarterly. With these programs you can use your computer to find which bank account pays more interest.

You may also wish to contribute regularly to an individual retirement account (IRA). With one of the programs in this chapter you can compute how much money you will have in your IRA at the time you retire. Another program allows you to see if an amount in a savings account is enough for you to retire with.

The last program in this chapter shows you how to put all the other programs together to produce a menu-controlled financial package.

FUTURE VALUE OF A DEPOSIT

Description

Have you ever wondered how quickly a savings account will grow? Would you like to compare the effect of different interest rates and compounding periods?

You can use this program to compute the future value of a deposit. You must specify the amount of the deposit, the number of times per year interest is compounded, the annual interest rate, and the number of years the funds will earn interest. The program assumes a constant interest rate.

Example

Tom deposits $5,000 in a four-year savings account. The bank pays interest at the rate of 12% per annum, compounded quarterly. Tom wishes to know how much his savings will grow in four years. Figure 2.1 shows the results of the computation on the screen. Figure 2.2 is the program listing for the Future Value program.

```
FUTURE VALUE OF A DEPOSIT

  INITIAL DEPOSIT ($)    5000
  PERIODS PER YEAR       4
  ANNUAL INT RATE (%)    12
  NUMBER OF YEARS        4

FUTURE VALUE           8023.53

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 2.1: Screen Display: Future Value of a Deposit

```
300  REM ----------------------------
310 N$ = "FUTURE VALUE OF A DEPOSIT"
320  REM ----------------------------
330 K = 1
340  GOSUB 7500: REM INITIALIZE
350  REM  ASK FOR PARAMETERS
360 Q1$ = "ENTER DATA:":Q2$ = ""
370 Q3$ = "INITIAL DEPOSIT ($)": GOSUB 8000
380 Q3$ = "PERIODS PER YEAR": GOSUB 8000
390 Q3$ = "ANNUAL INT RATE (%)": GOSUB 8000
400 Q3$ = "NUMBER OF YEARS": GOSUB 8000
410  REM  DO THE COMPUTATION
420  PRINT : PRINT "FUTURE VALUE"; TAB( 23);
430 A = PAR(1)
440  FOR I = 1 TO PAR(2) * PAR(4)
450 A = A + A * PAR(3) / 100 / PAR(2)
460  NEXT I
470 A =  INT (A * 100 + .5) / 100
480  PRINT A
490  REM  ASK FOR ANOTHER
500 Q2$ = "COMPUTE": GOSUB 4500
510  IF YN$ = "N" THEN  END
520  GOTO 300
```

Figure 2.2: Program Listing: Future Value of a Deposit

FUTURE VALUE OF A SERIES OF DEPOSITS

Description

This program calculates the future value of a series of regular deposits. For example, you can determine the future value of an IRA if you make regular annual contributions. You specify the amount of the payment, the number of payments made per year, the annual interest rate your funds earn, and the number of years you will make the payments. The program assumes that the compounding period and number of payments per year are the same, and that the annual interest rate is constant.

Example

Ann plans to deposit $500 in a savings account every month. Interest is earned at 6% per annum and is compounded monthly. How much money will be in her account in four years? Figure 2.3 shows the screen display for this example. The program listing is in Figure 2.4.

```
FUTURE VALUE OF A SERIES

  PAYMENT ($)              500
  PAYMENTS PER YEAR        12
  ANNUAL INT RATE (%)      6
  NUMBER OF YEARS          4

FUTURE VALUE             27048.92

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 2.3: Screen Display: Future Value of a Series of Deposits

```
530  REM --------------------------
540 N$ = "FUTURE VALUE OF A SERIES"
550  REM --------------------------
560 K = 1
570  GOSUB 7500: REM INITIALIZE
580  REM  ASK FOR PARAMETERS
590 Q1$ = "ENTER DATA:":Q2$ = ""
600 Q3$ = "PAYMENT ($)": GOSUB 8000
610 Q3$ = "PAYMENTS PER YEAR": GOSUB 8000
620 Q3$ = "ANNUAL INT RATE (%)": GOSUB 8000
630 Q3$ = "NUMBER OF YEARS": GOSUB 8000
640  REM  COMPUTE RESULT
650  PRINT : PRINT "FUTURE VALUE"; TAB( 23);
660 A = 0
670  FOR I = 1 TO PAR(2) * PAR(4)
680 A = PAR(1) + A + A * PAR(3) / 100 / PAR(2)
690  NEXT I
700 A =  INT (A * 100 + .5) / 100
710  PRINT A
720  REM  ASK FOR ANOTHER
730 Q2$ = "COMPUTE": GOSUB 4500
740  IF YN$ = "N" THEN  END
750  GOTO 530
```

Figure 2.4: Program Listing: Future Value of a Series of Deposits

PRESENT VALUE OF AN AMOUNT

Description

Assume that Bob owes you $5,000 in three years. He offers to pay you $4,000 today, and you want to know whether to accept his offer. The Present Value program uses an interest rate you specify to find the present value of a future amount. You should accept Bob's offer if the present value is less than $4,000.

To use this program you supply the future value, the annual interest rate at which your funds earn money, the number of compounding periods per year, and the number of years in the future that the value of the amount is known.

Example

Bill has a note for which he will receive $10,000 five years from now. Bill normally earns 10% interest on his funds, with interest being compounded quarterly. Bill would like to know the present value of his note. Figure 2.5 shows the screen display for the Present Value program; Figure 2.6 lists the Present Value program.

```
PRESENT VALUE OF AMOUNT

  FUTURE VALUE ($)        10000
  PERIODS PER YEAR        4
  ANNUAL INT RATE (%)     10
  NUMBER OF YEARS         5

PRESENT VALUE           6102.71

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 2.5: Screen Display: Present Value of an Amount

```
760  REM ------------------------
770 N$ = "PRESENT VALUE OF AMOUNT"
780  REM ------------------------
790 K = 1
800  GOSUB 7500: REM INITIALIZE
810  REM  ASK FOR PARAMETERS
820 Q1$ = "ENTER DATA:":Q2$ = ""
830 Q3$ = "FUTURE VALUE ($)": GOSUB 8000
840 Q3$ = "PERIODS PER YEAR": GOSUB 8000
850 Q3$ = "ANNUAL INT RATE (%)": GOSUB 8000
860 Q3$ = "NUMBER OF YEARS": GOSUB 8000
870  REM  DO THE COMPUTATION
880  PRINT : PRINT "PRESENT VALUE"; TAB( 23);
890 B = 1 + PAR(3) / 100 / PAR(2)
900 A = PAR(1) / B ^ (PAR(2) * PAR(4))
910 A =  INT (A * 100 + .5) / 100
920  PRINT A
930  REM  ASK FOR ANOTHER
940 Q2$ = "COMPUTE": GOSUB 4500
950  IF YN$ = "N" THEN  END
960  GOTO 760
```

Figure 2.6: Program Listing: Present Value of an Amount

PRESENT VALUE OF A SERIES OF PAYMENTS

Description

The previous example shows you how to find the present value of a future amount. With this program you can find the present value of a series of payments. For example, assume that Sally buys your car and pays you $50 a month for three years. After one year she offers you $900 in cash, and you want to know whether to accept her offer. You can use this program to find the answer to your question. You should accept if the present value is less than $900.

To use this program you specify the amount of the payment, the number of payments you receive per year, the number of years for which the payments will continue, and the annual interest rate at which your funds earn interest. We assume that the number of payments per year and the number of compounding periods are identical.

Example

Jackie receives $400 per month in payments. She can earn 10% interest on invested funds, and she wants to know the present value of the payments. The payments will continue for ten years. Figure 2.7 shows the screen display for this example, and Figure 2.8 is the complete program listing.

```
PRESENT VALUE OF A SERIES

  PAYMENT ($)              400
  PAYMENTS PER YEAR        12
  ANNUAL INT RATE (%)      10
  NUMBER OF YEARS          10

PRESENT VALUE          30268.46

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 2.7: Screen Display: Present Value of a Series of Payments

```
970  REM ---------------------------
980 N$ = "PRESENT VALUE OF A SERIES"
990  REM ---------------------------
1000 K = 1
1010  GOSUB 7500: REM INITIALIZE
1020  REM  ASK FOR PARAMETERS
1030 Q1$ = "ENTER DATA:":Q2$ = ""
1040 Q3$ = "PAYMENT ($)": GOSUB 8000
1050 Q3$ = "PAYMENTS PER YEAR": GOSUB 8000
1060 Q3$ = "ANNUAL INT RATE (%)": GOSUB 8000
1070 Q3$ = "NUMBER OF YEARS": GOSUB 8000
1080  REM  DO THE COMPUTATION
1090  PRINT : PRINT "PRESENT VALUE"; TAB( 23);
1100 A = 0
1110  FOR I = 1 TO PAR(2) * PAR(4)
1120 A = A + PAR(1) / ((1 + PAR(3) / 100 / PAR(2)) ^ I)
1130  NEXT I
1140 A =  INT (A * 100 + .5) / 100
1150  PRINT A
1160  REM  ASK FOR ANOTHER
1170 Q2$ = "COMPUTE": GOSUB 4500
1180  IF YN$ = "N" THEN  END
1190  GOTO 970
```

Figure 2.8: Program Listing: Present Value of a Series of Payments

PAYMENT REQUIRED FOR FUTURE SUM

Description

This program allows you to determine the regular payment necessary to produce a required amount of money in the future. For example, you can use this program to calculate what regular payments will be necessary to take a special vacation, replace a piece of equipment, or save for a college education. You must specify the future sum, the number of payments per year, the annual interest rate at which your funds earn interest, and the number of years to the time you need the money. The program assumes that the number of payments and the compounding periods are identical.

Example

Bob plans to make monthly payments into a college fund for his daughter. He will make the payments for 18 years, and he estimates that he will need $40,000. He can earn 8.5% interest on his college savings account. We show the screen display for this example in Figure 2.9 and the program listing in Figure 2.10.

```
PAYMENT REQD FOR FUTURE SUM

  FUTURE VALUE ($)        40000
  PAYMENTS PER YEAR       12
  ANNUAL INT RATE (%)     8.5
  NUMBER OF YEARS         18

PAYMENT                 78.85

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 2.9: Screen Display: Payment Required for Future Sum

```
1200  REM ------------------------------
1210 N$ = "PAYMENT REQD FOR FUTURE SUM"
1220  REM ------------------------------
1230 K = 1
1240  GOSUB 7500: REM INITIALIZE
1250  REM  ASK FOR PARAMETERS
1260 Q1$ = "ENTER DATA":Q2$ = ""
1270 Q3$ = "FUTURE VALUE ($)": GOSUB 8000
1280 Q3$ = "PAYMENTS PER YEAR": GOSUB 8000
1290 Q3$ = "ANNUAL INT RATE (%)": GOSUB 8000
1300 Q3$ = "NUMBER OF YEARS": GOSUB 8000
1310  REM  DO THE COMPUTATION
1320  PRINT : PRINT "PAYMENT"; TAB( 23);
1330 I = PAR(3) / 100 / PAR(2)
1340 Q = PAR(2) * PAR(4)
1350 A = PAR(1) * I / ((1 + I) ^ Q - 1)
1360 A =  INT (A * 100 + .5) / 100
1370  PRINT A
1380  REM  ASK FOR ANOTHER
1390 Q2$ = "COMPUTE": GOSUB 4500
1400  IF YN$ = "N" THEN  END
1410  GOTO 1200
```

Figure 2.10: Program Listing: Payment Required for Future Sum

WITHDRAWAL OF FUNDS

Description

You can use this program to determine how long you can make withdrawals from a bank account before the account runs out of funds. You must specify the initial deposit, the number of withdrawals per year, the interest rate at which the remaining funds earn interest, and the amount of the withdrawal. Again, the program assumes that the number of withdrawals per year and the number of compounding periods are identical.

If the withdrawal amount is less than the interest earned, your bank account will grow rather than run out of funds. The program then displays the message BANK BALANCE IS GROWING. Don't be concerned if this program takes a long time to finish running; the program executes as many times as necessary to find a zero bank balance.

Example

Jack and Jill Jones are ready to retire. They have $200,000 in their bank account, and the account earns 9% interest. Jack and Jill plan to live on $1,600 per month. How long will they be able to withdraw money? Figure 2.11 shows the screen display for this example; the program is listed in Figure 2.12.

```
WITHDRAWAL OF FUNDS

  INITIAL DEPOSIT ($)    200000
  WITHDWLS PER YEAR      12
  ANNUAL INT RATE (%)    9
  PAYMENT ($)            1600

PAYMENTS                 371
YEARS OF PAYMENTS        30.9166667

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 2.11: Screen Display: Withdrawal of Funds

```
1420  REM ----------------------
1430 N$ = "WITHDRAWAL OF FUNDS"
1440  REM ----------------------
1450 K = 1
1460  GOSUB 7500: REM INITIALIZE
1470  REM  ASK FOR PARAMETERS
1480 Q1$ = "ENTER DATA":Q2$ = ""
1490 Q3$ = "INITIAL DEPOSIT ($)": GOSUB 8000
1500 Q3$ = "WITHDWLS PER YEAR": GOSUB 8000
1510 Q3$ = "ANNUAL INT RATE (%)": GOSUB 8000
1520 Q3$ = "PAYMENT ($)": GOSUB 8000
1530  REM  DO THE COMPUTATION
1540  PRINT : PRINT "PAYMENTS"; TAB( 23);
1550 A = PAR(1):N = 1
1560 A = A + A * PAR(3) / 100 / PAR(2) - PAR(4)
1570  IF A < PAR(1) THEN 1610
1580  PRINT : PRINT "BANK BALANCE IS GROWING"
1590  PRINT  CHR$ (7)
1600  GOTO 1660
1610  IF A < PAR(4) THEN 1640
1620 N = N + 1
1630  GOTO 1560
1640  PRINT N
1650  PRINT "YEARS OF PAYMENTS"; TAB( 23);N / 12
1660  REM  ASK FOR ANOTHER
1670 Q2$ = "COMPUTE": GOSUB 4500
1680  IF YN$ = "N" THEN  END
1690  GOTO 1420
```

Figure 2.12: Program Listing: Withdrawal of Funds

INTEREST RATE EARNED

Description

The program in this example is useful when you want to determine the interest rate you earn when you know the beginning and ending values of your investment. For example, you can find the equivalent interest rate on the profit from a house sale. To use this program you specify the beginning and ending values of your funds, the number of compounding periods per year, and the number of years over which your funds have grown.

Example

Auntie Mame gave $5,000 to her broker to invest four years ago. She has just received $9,000 in return. She wishes to compute the equivalent interest rate she has earned, with interest compounded quarterly. Figures 2.13 and 2.14 show the screen display and program listing for this example.

```
INTEREST RATE EARNED

  AMOUNT INVESTED ($)    5000
  AMOUNT RETURNED ($)    9000
  PERIODS PER YEAR       4
  NUMBER OF YEARS        4

ANNUAL INT RATE (%)    14.97

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 2.13: Screen Display: Interest Rate Earned

```
1700  REM ----------------------
1710 N$ = "INTEREST RATE EARNED"
1720  REM ----------------------
1730 K = 1
1740  GOSUB 7500: REM INITIALIZE
1750  REM  ASK FOR PARAMETERS
1760 Q1$ = "ENTER DATA:":Q2$ = ""
1770 Q3$ = "AMOUNT INVESTED ($)": GOSUB 8000
1780 Q3$ = "AMOUNT RETURNED ($)": GOSUB 8000
1790 Q3$ = "PERIODS PER YEAR": GOSUB 8000
1800 Q3$ = "NUMBER OF YEARS": GOSUB 8000
1810  REM  DO THE COMPUTATION
1820  PRINT : PRINT "ANNUAL INT RATE (%)"; TAB( 23);
1830 Q = PAR(3) * PAR(4)
1840 I = ((PAR(2) / PAR(1)) ^ (1 / Q) - 1) * 100 * PAR(3)
1850 I =  INT (I * 100 + .5) / 100
1860  PRINT I
1870  REM  ASK FOR ANOTHER
1880 Q2$ = "COMPUTE": GOSUB 4500
1890  IF YN$ = "N" THEN  END
1900  GOTO 1700
```

Figure 2.14: Program Listing: Interest Rate Earned

NET PRESENT VALUE, UNEVEN CASH FLOWS

Description

Net present value (NPV) analysis is used by financial analysts to determine whether an investment is worthwhile. For an interest rate you specify, a positive NPV indicates that the investment is desirable.

This program is useful when you wish to determine the present value of a series of cash flows. The cash flows need not be the same in every year. You use this program by entering the initial investment you make and the annual interest rate at which your funds earn interest. The program then prompts you for each cash flow. You enter these cash flows as positive or negative numbers, depending on whether you receive or distribute funds. You signal the end of cash flow entries by typing 0.

Example

Tina can buy a note for $5,000. She will receive annual payments of $500, $1,500, $1,500, $1,500, and $1,000. Tina would like her funds to earn 12% annual interest. What is the net present value of her investment? The display in Figure 2.15 is negative, indicating that she is not earning 12%; thus she should not buy the note. A complete program listing is shown in Figure 2.16.

```
1910  REM ---------------------------------------
1920 N$ = "NET PRESENT VALUE, UNEVEN CASH FLOWS"
1930  REM ---------------------------------------
1940  DIM PV(100)
1950  GOSUB 7500: REM INITIALIZE
1960 K = 1
1970  REM  ASK FOR PARAMETERS
1980 Q1$ = "ENTER DATA:":Q2$ = ""
1990 Q3$ = "AMOUNT INVESTED ($)": GOSUB 8000
2000 Q3$ = "ANNUAL INT RATE (%)": GOSUB 8000
2010 Q1$ = "ENTER ANNUAL CASH FLOWS"
2020 Q2$ = "IN ORDER RECEIVED"
```

Figure 2.16: Program Listing: Net Present Value, Uneven Cash Flows (continues)

```
NET PRESENT VALUE, UNEVEN CASH FLOWS

  AMOUNT INVESTED ($)    5000
  ANNUAL INT RATE (%)    12


NET PRESENT VALUE       -769.41


     CASH FLOW 1:          500
     CASH FLOW 2:          1500
     CASH FLOW 3:          1500
     CASH FLOW 4:          1500
     CASH FLOW 5:          1000


CASH FLOW 6         ?0

---------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 2.15: Screen Display: Net Present Value, Uneven Cash Flows

```
2030 Q3$ = "TYPE 0 WHEN DONE"
2040  GOSUB 6000
2050  FOR I = 1 TO 100
2060  VTAB 18: HTAB 1: PRINT  SPC( 35)
2070  VTAB 18: HTAB 1
2080  PRINT "CASH FLOW ";I;"        ";
2090  INPUT PV(I)
2100  IF PV(I) = 0 THEN 2150
2110  VTAB I -  INT (I / 8) * 8 + 9: PRINT  SPC( 35)
2120  VTAB I -  INT (I / 8) * 8 + 9: HTAB 5
2130  PRINT "CASH FLOW ";I;":"; TAB( 25);PV(I)
2140  NEXT I
2150 NPV =  - PAR(1)
2160  FOR I = 1 TO 100
2170 NPV = NPV + PV(I) / (1 + PAR(2) / 100) ^ I
2180  NEXT I
2190  VTAB 7: HTAB 1
2200  PRINT "NET PRESENT VALUE"; TAB( 23);
2210 A =  INT (NPV * 100 + .5) / 100
2220  PRINT A
2230  FOR I = 1 TO 100
2240 PV(I) = 0
2250  NEXT I
2260  REM  ASK FOR ANOTHER
2270 Q2$ = "COMPUTE": GOSUB 4500
2280  IF YN$ = "N" THEN  END
2290  GOTO 1950
```

Figure 2.16: Program Listing: Net Present Value, Uneven Cash Flows

TIME TO DOUBLE

Description

This program computes the time it will take for the funds you invest to double. To use the program you enter the annual interest rate and number of compounding periods per year.

Example

Fred's bank compounds interest four times yearly and pays 12% interest per annum. How many years will it take Fred to double a $5,000 investment? We display the results of this program in Figure 2.17; a program listing is shown in Figure 2.18.

```
TIME TO DOUBLE

  PERIODS PER YEAR       4
  ANNUAL INT RATE (%)    12

TIME TO DOUBLE (YRS)     5.86

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 2.17: Screen Display: Time to Double

```
2300  REM ----------------
2310 N$ = "TIME TO DOUBLE"
2320  REM ----------------
2330  REM  INITIALIZE
2340 K = 1: GOSUB 7500
2350  REM  ASK FOR PARAMETERS
2360 Q1$ = "ENTER DATA:":Q2$ = ""
2370 Q3$ = "PERIODS PER YEAR": GOSUB 8000
2380 Q3$ = "ANNUAL INT RATE (%)": GOSUB 8000
2390  REM  DO THE COMPUTATION
2400  PRINT : PRINT "TIME TO DOUBLE (YRS)"; TAB( 25);
2410 I = PAR(2) / 100 / PAR(1)
2420 Y =  LOG (2) / (PAR(1) *  LOG (1 + I))
2430 Y =  INT (Y * 100 + .5) / 100
2440  PRINT Y
2450  REM  ASK FOR ANOTHER
2460 Q2$ = "COMPUTE": GOSUB 4500
2470  IF YN$ = "N" THEN  END
2480  GOTO 2300
```

Figure 2.18: Program Listing: Time to Double

EQUIVALENT INTEREST RATE

Description

Many of the programs in this chapter assume that the number of compounding periods and the number of payments are identical. This program allows you to determine an equivalent interest rate when the two periods are not the same. To use the program, enter the annual interest rate, the number of compounding periods, and the number of payments you want to make per year (desired periods). The program then computes an equivalent interest rate that you can use.

Example

You make deposits monthly into an account, and you wish to determine the future value of the deposits. The interest on this account is 12% per annum, compounded quarterly. As shown in Figure 2.19, the equivalent interest rate is 11.88%. You should use this figure and then run the Future Value of a Series of Deposits program to compute your final answer. The program listing for this example is given in Figure 2.20.

```
EQUIVALENT INTEREST RATE

  ANNUAL INT RATE (%)    12
  AVAILABLE PERIODS      4
  DESIRED PERIODS        12

EQUIV INT RATE           11.88

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 2.19: Screen Display: Equivalent Interest Rate

```
2490  REM ------------------------
2500 N$ = "EQUIVALENT INTEREST RATE"
2510  REM ------------------------
2520 K = 1
2530  GOSUB 7500: REM INITIALIZE
2540  REM  ASK FOR PARAMETERS
2550 Q1$ = "ENTER DATA:":Q2$ = ""
2560 Q3$ = "ANNUAL INT RATE (%)": GOSUB 8000
2570 Q3$ = "AVAILABLE PERIODS": GOSUB 8000
2580 Q3$ = "DESIRED PERIODS": GOSUB 8000
2590 I = PAR(1):Q2 = PAR(2):Q1 = PAR(3)
2600 IEQ = Q1 * (1 + I / 100 / Q2) ^ (Q2 / Q1) - Q1
2610  PRINT : PRINT "EQUIV INT RATE"; TAB( 25);
2620 IEQ =  INT (IEQ * 10000 + .5) / 100
2630  PRINT IEQ
2640  REM  ASK FOR ANOTHER
2650 Q2$ = "COMPUTE": GOSUB 4500
2660  IF YN$ = "N" THEN  END
2670  GOTO 2490
```

Figure 2.20: Program Listing: Equivalent Interest Rate

FINANCIAL PROGRAMS MENU

All of the programs in this chapter have been entered with distinct program line numbers. If you save all the programs together in one diskette file, you can use the menu program we develop here to run any of these programs. The menu is set up so that you press a single key to select the program you want to run.

Remember, when these programs are used with the menu, you must replace the END instruction at the end of every program with a RETURN instruction. You should enter the menu program by first loading your financial programs, then keying in the menu program. Then save the programs and menu with the SAVE command. The menu is shown in Figure 2.21. Figure 2.22 shows the menu selection program.

After you have run the financial programs you need, you can return to BASIC by pressing the reset key—the cursor will appear on a blank line, and you can then load another set of programs.

```
1= FUTURE VALUE OF A DEPOSIT
2= FUTURE VALUE OF A SERIES
3= PRESENT VALUE OF AMOUNT
4= PRESENT VALUE OF A SERIES
5= PAYMENT REQD FOR FUTURE SUM
6= WITHDRAWAL OF FUNDS
7= INTEREST RATE EARNED
8= NET PRESENT VALUE, UNEVEN CASH FLOWS
9= TIME TO DOUBLE
0= EQUIVALENT INTEREST RATE

----------------------------------------

CHOOSE PROGRAM:
```

Figure 2.21: Screen Display: Financial Programs Menu

```
100  REM --------------------
110 N$ = "FINANCIAL PROGRAMS"
120  REM --------------------
130  GOSUB 7500: REM INITIALIZE
140  REM  SET UP MENU ARRAY
150 X$(1) = "FUTURE VALUE OF A DEPOSIT"
160 X$(2) = "FUTURE VALUE OF A SERIES"
170 X$(3) = "PRESENT VALUE OF AMOUNT"
180 X$(4) = "PRESENT VALUE OF A SERIES"
190 X$(5) = "PAYMENT REQD FOR FUTURE SUM"
200 X$(6) = "WITHDRAWAL OF FUNDS"
210 X$(7) = "INTEREST RATE EARNED"
220 X$(8) = "NET PRESENT VALUE, UNEVEN CASH FLOWS"
230 X$(9) = "TIME TO DOUBLE"
240 X$(10) = "EQUIVALENT INTEREST RATE"
250  REM  DISPLAY MENU
260 N = 10: GOSUB 8500
270  ON X GOSUB 300,530,760,970,1200,1420,1700,1910,2300,2490
280  GOTO 100
```

Figure 2.22: Program Listing: Financial Programs Menu

3 USEFUL BUSINESS PROGRAMS

This chapter presents ten helpful business programs. These programs enable you to prepare depreciation schedules, find the economic ordering quantity, and solve many other business problems.

For example, you may be starting a business or running a charity event; the Break-Even Point program will calculate how many items or tickets you must sell to produce a profit. Whether you are a salesperson or employ a salesperson, the Salesperson's Commission program allows you to find the commission due for a given sales volume. The Wages with Overtime program makes it simple to calculate the salary due when overtime is worked. This program adds base pay and overtime pay to find the total salary due. We also provide an executive decision making program, which can be used when all else fails.

Remember to save these programs as you enter them and to merge the central subroutines prior to running the programs.

STRAIGHT-LINE DEPRECIATION

Description

This program computes a yearly depreciation schedule based on the straight-line method. To depreciate an asset, you specify the present book value, the estimated salvage value, and the life of the asset in years. If your asset has a life longer than ten years, the program displays the schedule for the first ten years and then pauses. When you are ready to review years 11–20, you simply press any key, and the schedule for those years is displayed.

Example

Let's set up a depreciation schedule for an office copier. We presently value the copier at $5,000. We expect to use the copier for six years and then sell it for $1,000. Figure 3.1 shows the screen display of the depreciation schedule. Figure 3.2 is a listing of the depreciation program.

```
STRAIGHT-LINE DEPRECIATION

  PRES. BOOK VALUE ($)   5000
  SALVAGE VALUE ($)      1000
  LIFE (YRS)             6

YEAR   DEPREC($)      VALUE($)

 1       666.67        4333.33
 2       666.67        3666.66
 3       666.67        2999.99
 4       666.67        2333.32
 5       666.67        1666.65
 6       666.67        999.98


----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 3.1: Screen Display: Straight-Line Depreciation

```
310  REM ----------------------------
320 N$ = "STRAIGHT-LINE DEPRECIATION"
330  REM ----------------------------
340 K = 1
350  GOSUB 7500: REM INITIALIZE
360  REM  GET INPUT PARAMETERS
370 Q1$ = "ENTER DATA:":Q2$ = ""
380 Q3$ = "PRES. BOOK VALUE ($)": GOSUB 8000
390 Q3$ = "SALVAGE VALUE ($)": GOSUB 8000
400 Q3$ = "LIFE (YRS)": GOSUB 8000
410  REM  PRINT DEPREC. SCHED.
420 D = (PAR(1) - PAR(2)) / PAR(3)
430 D =  INT (D * 100 + .5) / 100
440  PRINT
450  PRINT "YEAR    DEPREC($)      VALUE($)"
460  FOR A = 1 TO PAR(3)
470  VTAB A -  INT ((A - 1) / 10) * 10 + 8
480  HTAB 1: PRINT  SPC( 35): HTAB 1
490  VTAB A -  INT ((A - 1) / 10) * 10 + 8
500  PRINT " ";A; TAB( 9);D; TAB( 23);PAR(1) - D * A
510  IF A / 10 =  INT (A / 10) THEN  GOSUB 9000
520  NEXT A
530  REM  ASK FOR ANOTHER
540 Q2$ = "COMPUTE": GOSUB 4500
550  IF YN$ = "N" THEN  END
560  GOTO 310
```

Figure 3.2: Program Listing: Straight-Line Depreciation

DECLINING-BALANCE DEPRECIATION

Description

This program makes it easy to calculate depreciation schedules based on the declining-balance method, which is an accelerated method of depreciation. Accelerated depreciation methods are often used to provide more favorable tax treatment than the straight-line method provides. The declining-balance method multiplies the remaining book value of an asset by an acceleration factor and then divides by the life of the asset in years. You must specify the initial book value of the asset, its useful life, and the acceleration factor. Typical acceleration factors are 150%, 175%, and 200%; the declining-balance method ignores the salvage value.

Example

Let's now depreciate the office copier in the previous example by the declining-balance method. In this example we use an acceleration factor of 150%. The depreciation schedule appears in Figure 3.3 and the program listing in Figure 3.4.

```
DECLINING-BALANCE DEPRECIATION

  PRES. BOOK VALUE ($)   5000
  LIFE (YRS)             6
  % ACCEL. DEPREC.       150

YEAR     DEPREC($)     VALUE($)

 1         1250          3750
 2         937.5         2812.5
 3         703.13        2109.37
 4         527.34        1582.03
 5         395.51        1186.52
 6         296.63        889.89




----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 3.3: Screen Display: Declining-Balance Depreciation

```
570  REM --------------------------------
580 N$ = "DECLINING-BALANCE DEPRECIATION"
590  REM --------------------------------
600 K = 1: GOSUB 7500
610  REM  GET INPUT PARAMETERS
620 Q1$ = "ENTER DATA:":Q2$ = ""
630 Q3$ = "PRES. BOOK VALUE ($)": GOSUB 8000
640 Q3$ = "LIFE (YRS)": GOSUB 8000
650 Q3$ = "% ACCEL. DEPREC.": GOSUB 8000
660  REM  PRINT DEPREC. SCHEDULE
670 RV = PAR(1)
680  PRINT
690  PRINT "YEAR     DEPREC($)     VALUE($)"
700  FOR A = 1 TO PAR(2)
710 D = RV * PAR(3) / 100 / PAR(2)
720 D =  INT (D * 100 + .5) / 100
730  VTAB A -  INT ((A - 1) / 10) * 10 + 8
740  HTAB 1: PRINT  SPC( 35): HTAB 1
750  VTAB A -  INT ((A - 1) / 10) * 10 + 8
760  PRINT " ";A; TAB( 10);D; TAB( 23);RV - D
770 RV = RV - D
780 RV =  INT (RV * 100 + .5) / 100
790  IF A / 10 =  INT (A / 10) THEN  GOSUB 9000
800  NEXT A
810  REM  ASK FOR ANOTHER
820 Q2$ = "COMPUTE": GOSUB 4500
830  IF YN$ = "N" THEN  END
840  GOTO 570
```

Figure 3.4: Program Listing: Declining-Balance Depreciation

SUM-OF-YEARS'-DIGITS DEPRECIATION

Description

We now show a second method of accelerated depreciation, the sum of the years' digits. This depreciation method multiplies the remaining book value by the ratio of the number of remaining years to the sum of the years of life. If the life of the asset is five years, the sum of the years of life is

$$5 + 4 + 3 + 2 + 1 = 15$$

You again specify the present book value, salvage value, and life of the asset. Your computer simplifies the otherwise tedious calculation.

Example

Let us produce a sum-of-the-years'-digits depreciation schedule for the office copier in the prior examples. The schedule is shown in Figure 3.5; the program listing appears in Figure 3.6.

```
850  REM ----------------------------------
860 N$ = "SUM-OF-YEARS'-DIGITS DEPRECIATION"
870  REM ----------------------------------
880  GOSUB 7500:K = 1
890  REM  GET INPUT PARAMETERS
900 Q1$ = "ENTER DATA:":Q2$ = ""
910 Q3$ = "PRES. BOOK VALUE ($)": GOSUB 8000
920 Q3$ = "SALVAGE VALUE ($)": GOSUB 8000
930 Q3$ = "LIFE (YRS)": GOSUB 8000
940  REM  PRINT DEPREC. SCHED.
950 D = PAR(1) - PAR(2)
960 RV = PAR(1)
```

Figure 3.6: Program Listing: Sum-of-Years'-Digits Depreciation (continues)

```
SUM-OF-YEARS'-DIGITS DEPRECIATION

  PRES. BOOK VALUE ($)   5000
  SALVAGE VALUE ($)      1000
  LIFE (YRS)             6

YEAR     DEPREC($)     VALUE($)

 1        1142.86       3857.14
 2        952.38        2904.76
 3        761.9         2142.86
 4        571.43        1571.43
 5        380.95        1190.48
 6        190.48        1000




----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 3.5: Screen Display: Sum-of-Years'-Digits Depreciation

```
970  PRINT
980  PRINT "YEAR     DEPREC($)     VALUE($)"
990  FOR A = 1 TO PAR(3)
1000 B = (PAR(3) + 1) / 2
1010 D1 = D * (PAR(3) + 1 - A) / (PAR(3) * B)
1020 D1 =  INT (D1 * 100 + .5) / 100
1030  VTAB A -  INT ((A - 1) / 10) * 10 + 8
1040  HTAB 1: PRINT  SPC( 38): HTAB 1
1050  VTAB A -  INT ((A - 1) / 10) * 10 + 8
1060  PRINT " ";A; TAB( 10);D1; TAB( 23);RV - D1
1070 RV = RV - D1
1080  IF A / 10 =  INT (A / 10) THEN  GOSUB 9000
1090  NEXT A
1100  REM  ASK FOR ANOTHER
1110 Q2$ = "COMPUTE": GOSUB 4500
1120  IF YN$ = "N" THEN  END
1130  GOTO 850
```

Figure 3.6: Program Listing: Sum-of-Years'-Digits Depreciation

BREAK-EVEN POINT

Description

This program allows you to find the number of items you must sell for a business or product line to break even. To use this program you must specify the fixed costs of the business (overhead), the cost per unit to manufacture or buy your product, and the sales price per unit.

Example

Harry buys novelty items for $3.00 each and plans to sell them for $6.00 each through mail order. If Harry plans to spend $1,500 to promote and sell the items, how many units must he sell to break even? The screen display for this example is shown in Figure 3.7; the program listing appears in Figure 3.8.

```
BREAK-EVEN POINT

  FIXED COSTS ($)          1500
  COST PER UNIT ($)        3
  SALES PRICE ($)          6

BREAK-EVEN POINT -> 500 UNITS

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 3.7: Screen Display: Break-Even Point

```
1140  REM ------------------
1150 N$ = "BREAK-EVEN POINT"
1160  REM ------------------
1170  GOSUB 7500:K = 1
1180  REM  GET INPUT PARAMETERS
1190 Q1$ = "ENTER DATA:":Q2$ = ""
1200 Q3$ = "FIXED COSTS ($)": GOSUB 8000
1210 Q3$ = "COST PER UNIT ($)": GOSUB 8000
1220 Q3$ = "SALES PRICE ($)": GOSUB 8000
1230  PRINT
1240 US =  INT (PAR(1) / (PAR(3) - PAR(2)) * 100 + .5) / 100
1250  PRINT "BREAK-EVEN POINT -> ";US;" UNITS"
1260  REM  ASK FOR ANOTHER
1270 Q2$ = "COMPUTE": GOSUB 4500
1280  IF YN$ = "N" THEN  END
1290  GOTO 1140
```

Figure 3.8: Program Listing: Break-Even Point

ECONOMIC ORDERING QUANTITY

Description

Operations researchers have developed a method to compute the most economic quantity of items for a manufacturer to order. This program does the computation for you. To use this program, specify the cost to place a purchase order, the number of units you use annually, and the annual carrying cost per unit. Carrying cost is defined as the interest rate your funds earn multiplied by the purchase price per item.

Example

Ray is the production manager for a small manufacturing company. He wants to calculate the economic ordering quantity for pumps that are used in the company's products. The company uses 15,000 pumps annually. The annual carrying cost of the pumps is $5 each, and a purchase order costs $75. What is the economic ordering quantity? Figure 3.9 shows the result of Ray's analysis. Figure 3.10 displays the program listing.

```
ECONOMIC ORDERING QUANTITY

  COST TO ORDER ($)      75
  ANNUAL UNITS USED      15000
  UNIT CARRYING COST ($)5

   EOQ=670 UNITS

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 3.9: Screen Display: Economic Ordering Quantity

```
1300  REM ----------------------------
1310 N$ = "ECONOMIC ORDERING QUANTITY"
1320  REM ----------------------------
1330  GOSUB 7500:K = 1
1340  REM  GET INPUT PARAMETERS
1350 Q1$ = "ENTER DATA:":Q2$ = ""
1360 Q3$ = "COST TO ORDER ($)": GOSUB 8000
1370 Q3$ = "ANNUAL UNITS USED": GOSUB 8000
1380 Q3$ = "UNIT CARRYING COST ($)": GOSUB 8000
1390  REM  COMPUTE AND PRINT
1400  PRINT
1410 EOQ =  INT ( SQR (2 * PAR(1) * PAR(2) / PAR(3)))
1420  PRINT "   EOQ=";EOQ;" UNITS"
1430  REM  ASK FOR ANOTHER
1440 Q2$ = "COMPUTE": GOSUB 4500
1450  IF YN$ = "N" THEN  END
1460  GOTO 1300
```

Figure 3.10: Program Listing: Economic Ordering Quantity

SALES PRICE WITH DISCOUNT

Description

Here is a program that enables you to compute the total sales price, with tax, of an item that is selling at a discount. To use this program you specify the retail price, discount percentage, and sales tax rate.

Example

Betty wants to buy a television set that is being sold at a 15% discount. The retail price of the television set is $350 and the sales tax is 6.5%. How much will Betty have to pay? Figure 3.11 shows the screen display for this example. Figure 3.12 shows the listing for the Sales Price with Discount program.

```
SALES PRICE WITH DISCOUNT

  RETAIL PRICE ($)        350
  DISCOUNT (%)            15
  SALES TAX (%)           6.5

  PRICE                   350
  DISCOUNT                52.5
  TAX                     19.34

TOTAL                     316.84

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 3.11: Screen Display: Sales Price with Discount

```
1470  REM ---------------------------
1480 N$ = "SALES PRICE WITH DISCOUNT"
1490  REM ---------------------------
1500  GOSUB 7500:K = 1
1510  REM  GET INPUT PARAMETERS
1520 Q1$ = "ENTER DATA:":Q2$ = ""
1530 Q3$ = "RETAIL PRICE ($)": GOSUB 8000
1540 Q3$ = "DISCOUNT (%)": GOSUB 8000
1550 Q3$ = "SALES TAX (%)": GOSUB 8000
1560  REM  COMPUTE AND PRINT
1570  PRINT
1580  PRINT "  PRICE"; TAB( 25);PAR(1)
1590 D = PAR(1) * PAR(2) / 100
1600 D =  INT (D * 100 + .5) / 100
1610  PRINT "  DISCOUNT"; TAB( 25);D
1620 T = (PAR(1) - D) * PAR(3) / 100
1630 T =  INT (T * 100 + .5) / 100
1640  PRINT "  TAX"; TAB( 25);T
1650  PRINT
1660  PRINT "TOTAL"; TAB( 25);PAR(1) - D + T
1670  REM  ASK FOR ANOTHER
1680 Q2$ = "COMPUTE": GOSUB 4500
1690  IF YN$ = "N" THEN  END
1700  GOTO 1470
```

Figure 3.12: Program Listing: Sales Price with Discount

WEIGHTED AVERAGE

Description

There are many times when you will need to compute the weighted average of a number of items. Weighted averages can be used to find the average cost of stocks when you buy shares of stock at different times and different values. Weighted averages are also one way of finding the value of a business's inventory when the same item is bought at different prices.

This program rapidly computes the weighted average of a series of quantities. You use the program by entering the cost per unit followed by the number of units purchased for each item. You end the entry process by typing 0 for both the unit value and the number of units.

Example

Lloyd has been buying stock in a rapidly appreciating electronics company. He would like to calculate his average cost of shares; he has purchased four shares at $10 each, four shares at $40 each, and two shares at $60 each. Figures 3.13 and 3.14 show the result of the calculation and the program listing.

```
1710  REM ------------------
1720 N$ = "WEIGHTED AVERAGE"
1730  REM ------------------
1740  DIM D(100,1)
1750  REM  GET INPUT PARAMETERS
1760  GOSUB 7500:K = 1
1770 Q1$ = "ENTER DATA: 0 UNITS WHEN DONE"
1780 Q2$ = "":Q3$ = "": GOSUB 6000
1790 ND = 0
1800  VTAB 22: HTAB 1: PRINT  SPC( 78)
1810  VTAB 22: HTAB 1: INPUT "UNIT VALUE? ";D(ND,0)
```

Figure 3.14: Program Listing: Weighted Average (continues)

```
WEIGHTED AVERAGE


   UNIT VALUE: 10        UNITS: 4
   UNIT VALUE: 40        UNITS: 4
   UNIT VALUE: 60        UNITS: 2




  WEIGHTED AVERAGE IS: 32

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 3.13: Screen Display: Weighted Average

```
1820  INPUT "UNITS?       ";D(ND,1)
1830  IF D(ND,1) = 0 THEN 1920
1840  VTAB ND - INT (ND / 10) * 10 + 6
1850  PRINT  SPC( 39)
1860  VTAB ND - INT (ND / 10) * 10 + 6: HTAB 1
1870  PRINT  SPC( 38): HTAB 1
1880  VTAB ND - INT (ND / 10) * 10 + 6
1890  PRINT "   UNIT VALUE: ";D(ND,0);
1900  PRINT  TAB( 25);"UNITS: ";D(ND,1)
1910 ND = ND + 1: GOTO 1800
1920  REM  NOW COMPUTE AND PRINT
1930 AVE = 0:U = 0
1940  FOR I = 0 TO ND - 1
1950 AVE = AVE + D(I,0) * D(I,1)
1960 U = U + D(I,1)
1970  NEXT I
1980 AVE = AVE / U
1990  VTAB 17: HTAB 3
2000  PRINT "WEIGHTED AVERAGE IS: ";AVE
2010  REM  ASK FOR ANOTHER
2020 Q2$ = "COMPUTE": GOSUB 4500
2030  IF YN$ = "N" THEN  END
2040  GOTO 1760
```

Figure 3.14: Program Listing: Weighted Average

SALESPERSON'S COMMISSION

Description

We will now enter a program to calculate the commission due to a salesperson. You specify the salesperson's monthly draw, monthly sales volume, and commission percentage. A negative result indicates that the salesperson has not sold his quota.

Example

Sam Spade receives a monthly draw of $1,200. He earns 6% commission on the sales he makes as a furniture salesman. What commission is he owed in a month during which he sells $25,000 worth of furniture? Figures 3.15 and 3.16 are the screen display and the program listing for the Salesperson's Commission program.

```
SALESPERSON'S COMMISSION

  MONTHLY DRAW ($)        1200
  MONTHLY SALES ($)       25000
  COMMISSION (%)          6

COMMISSION EARNED:        1500
DUE (LESS DRAW):          300

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 3.15: Screen Display: Salesperson's Commission

```
2050  REM ------------------------
2060 N$ = "SALESPERSON'S COMMISSION"
2070  REM ------------------------
2080  GOSUB 7500:K = 1
2090  REM  GET INPUT PARAMETERS
2100 Q1$ = "ENTER DATA:":Q2$ = ""
2110 Q3$ = "MONTHLY DRAW ($)": GOSUB 8000
2120 Q3$ = "MONTHLY SALES ($)": GOSUB 8000
2130 Q3$ = "COMMISSION (%)": GOSUB 8000
2140  REM  COMPUTE AND PRINT
2150  PRINT
2160  PRINT "COMMISSION EARNED:"; TAB( 25);
2170  PRINT  INT (PAR(2) * PAR(3) + .5) / 100
2180  PRINT "DUE (LESS DRAW):"; TAB( 25);
2190 D = PAR(2) * PAR(3) / 100 - PAR(1)
2200  PRINT  INT (D * 100 + .5) / 100
2210  REM  ASK FOR ANOTHER
2220 Q2$ = "COMPUTE": GOSUB 4500
2230  IF YN$ = "N" THEN  END
2240  GOTO 2050
```

Figure 3.16: Program Listing: Salesperson's Commission

WAGES WITH OVERTIME

Description

This program makes it easy to calculate the total wages due an employee who earns overtime pay at a premium rate. You enter the employee's base hourly pay, the overtime pay factor, and the number of base and overtime hours that the employee worked.

Example

Toni earns $10.00 an hour as a machine operator. During a very busy week she works her regular 40 hours plus 12 hours at overtime pay. You wish to compute her total wages, with overtime paid at time and one half. Figures 3.17 and 3.18 show the results of the overtime computation and the program listing.

```
WAGES WITH OVERTIME

  BASE RATE ($/HR)          10
  OT PAY FACTOR             1.5
  BASE HOURS WORKED         40
  OT HOURS WORKED           12

   BASE PAY                 400
   OT PAY                   180
   TOTAL PAY                580

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 3.17: Screen Display: Wages with Overtime

```
2250  REM ---------------------
2260 N$ = "WAGES WITH OVERTIME"
2270  REM ---------------------
2280  GOSUB 7500:K = 1
2290  REM  GET INPUT PARAMETERS
2300 Q1$ = "ENTER DATA:":Q2$ = ""
2310 Q3$ = "BASE RATE ($/HR)": GOSUB 8000
2320 Q3$ = "OT PAY FACTOR": GOSUB 8000
2330 Q3$ = "BASE HOURS WORKED": GOSUB 8000
2340 Q3$ = "OT HOURS WORKED": GOSUB 8000
2350  REM  COMPUTE AND PRINT
2360  PRINT
2370 D =  INT (PAR(1) * PAR(3) * 100 + .5) / 100
2380  PRINT "   BASE PAY"; TAB( 25);D
2390 D = PAR(1) * PAR(2) * PAR(4)
2400 D =  INT (D * 100 + .5) / 100
2410  PRINT "   OT PAY"; TAB( 25);D
2420 D = PAR(1) * PAR(3) + PAR(1) * PAR(2) * PAR(4)
2430 D =  INT (D * 100 + .5) / 100
2440  PRINT "   TOTAL PAY"; TAB( 25);D
2450  REM  ASK FOR ANOTHER
2460 Q2$ = "COMPUTE": GOSUB 4500
2470  IF YN$ = "N" THEN  END
2480  GOTO 2250
```

Figure 3.18: Program Listing: Wages with Overtime

EXECUTIVE DECISION MAKER

Description

When all else fails, you can use this program to assist you in making key business decisions. Simply type your question into the computer. The program uses advanced decision theory to answer your question. (This program is fun to run at a cocktail party or other social gatherings.)

Example

Ed needs to know whether a high-risk project will succeed. Because his staff members cannot provide him with a definite recommendation, he asks his computer, as shown in Figure 3.19. The listing for this program is in Figure 3.20.

```
EXECUTIVE DECISION MAKER

PLEASE TYPE YOUR QUESTION

=>WILL THIS PROJECT SUCCEED?
YOU'D BETTER BELIEVE IT









----------------------------------------
WOULD YOU LIKE TO
ASK A QUESTION
AGAIN? (Y OR N)
```

Figure 3.19: Screen Display: Executive Decision Maker

```
2490  REM  -------------------------
2500 N$ = "EXECUTIVE DECISION MAKER"
2510  REM  -------------------------
2520  GOSUB 7500:K = 1
2530  PRINT "PLEASE TYPE YOUR QUESTION"
2540  PRINT
2550  INPUT "=>";Q$:Y =  LEN (Q$)
2560  IF Y < 6 THEN 2580
2570 Y =  INT (Y / 2): GOTO 2560
2580  ON Y GOTO 2590,2600,2610,2620,2630
2590  PRINT "NOT ON YOUR LIFE": GOTO 2640
2600  PRINT "GO FOR IT!!!": GOTO 2640
2610  PRINT "YOU'D BETTER BELIEVE IT": GOTO 2640
2620  PRINT "FLIP A COIN": GOTO 2640
2630  PRINT "MAKE YOUR BREAK- QUICKLY"
2640  REM  ASK FOR ANOTHER
2650 Q2$ = "ASK A QUESTION": GOSUB 4500
2660  IF YN$ = "N" THEN  END
2670  GOTO 2490
```

Figure 3.20: Program Listing: Executive Decision Maker

BUSINESS PROGRAMS MENU

If you save all the programs in this chapter in one diskette file, you can add the menu program shown here to run the other programs. You select the programs by using the number keys of your computer. Remember to replace the END instruction in each program with a RETURN command. Review the menu entry instructions in Chapter 2 prior to creating your menu. Figures 3.21 and 3.22 display an image of the menu and the menu selection program.

```
1= STRAIGHT-LINE DEPRECIATION
2= DECLINING-BALANCE DEPRECIATION
3= SUM-OF-YEARS'-DIGITS DEPRECIATION
4= BREAK-EVEN POINT
5= ECONOMIC ORDERING QUANTITY
6= SALES PRICE WITH DISCOUNT
7= WEIGHTED AVERAGE
8= SALESPERSON'S COMMISSION
9= WAGES WITH OVERTIME
0= EXECUTIVE DECISION MAKER

----------------------------------------

CHOOSE PROGRAM:
```

Figure 3.21: Screen Display: Business Programs Menu

```
100  REM --------------------
110 N$ = "BUSINESS PROGRAMS"
120  REM --------------------
130  GOSUB 7500: REM INITIALIZE
140  FOR X = 1 TO 1000: NEXT
150  REM  SET UP MENU ARRAY
160 X$(1) = "STRAIGHT-LINE DEPRECIATION"
170 X$(2) = "DECLINING-BALANCE DEPRECIATION"
180 X$(3) = "SUM-OF-YEARS'-DIGITS DEPRECIATION"
190 X$(4) = "BREAK-EVEN POINT"
200 X$(5) = "ECONOMIC ORDERING QUANTITY"
210 X$(6) = "SALES PRICE WITH DISCOUNT"
220 X$(7) = "WEIGHTED AVERAGE"
230 X$(8) = "SALESPERSON'S COMMISSION"
240 X$(9) = "WAGES WITH OVERTIME"
250 X$(10) = "EXECUTIVE DECISION MAKER"
260  REM  DISPLAY MENU
270 N = 10: GOSUB 8500
280  ON X GOSUB 310,570,850,1140,1300,1470,1710,2050,2250,2490
290  GOTO 100
```

Figure 3.22: Program Listing: Business Programs Menu

4 REAL ESTATE PROGRAMS

In this chapter we present eight useful real estate programs. These programs allow you to set up mortgage payment schedules and perform many other practical computations.

Two programs in this chapter are especially interesting. You can use the Affordable House Price program to find the sales price of a home that a bank will finance. This program is helpful both in finding a price range you can afford and in negotiating a price should you decide to purchase. Another program you will find particularly enlightening is the Accelerated Payments program. A small increase in your monthly payment can drastically reduce the total interest you pay on a home. Take a look at Figure 4.8, and you'll see what we mean!

As in the other chapters, we provide a menu selection program that you can use to create a set of real estate programs.

REAL ESTATE SUBROUTINES

The programs in this chapter make use of three common subroutines. You should first enter these subroutines and save them on disk. When you are ready to enter a program, load the common subroutines. Then type in the program and save the subroutines and program together as a new disk file. (Merge SUBLIB with this file prior to running your program.) You should enter all the programs in the chapter on the same disk file with these subroutines. The common subroutines are shown in Figure 4.1. These routines are used to get input for a payment calculation, calculate a payment, and compute the remaining balance of a loan.

```
260  REM ---------------------
270  REM  ROUTINE TO CALC PMT
280  REM ---------------------
290  REM  CALLING PARAMETERS:
300  REM   IN=ANNUAL INT. RATE
310  REM   YR=NO. OF YEARS
320  REM   AM=AMOUNT OF LOAN
330  REM
340 N1 = 12 * YR:I1 = IN / 100 / 12:V = 1 / (1 + I1)
350 P = AM * I1 / (1 - V ^ N1)
360  RETURN
370  REM ----------------------
380  REM   INPUT FOR PMT CALC
390  REM ----------------------
400 Q1$ = "SPECIFY PARAMETERS"
410 Q2$ = ""
420 Q3$ = "AMOUNT BORROWED ($)": GOSUB 8000
430 Q3$ = "ANNUAL INT RATE (%)": GOSUB 8000
440 Q3$ = "TERM OF LOAN (YRS)": GOSUB 8000
450 AM = PAR(1)
460 IN = PAR(2)
470 YR = PAR(3)
480  RETURN
490  REM ----------------------
500  REM   CALC REMAINING BAL
510  REM ----------------------
520  REM  CALLING PARAMETERS:
530  REM   P=PAYMENT
540  REM   IN=ANNUAL INT. RATE
550  REM   N=PAYMENT NUMBER
560  REM   AM=AMOUNT OF LOAN
570  FOR I = 1 TO N
580 AM = AM - P + IN / 12 / 100 * AM
590  NEXT I
600  RETURN
```

Figure 4.1: Program Listing: Real Estate Subroutines

MONTHLY PAYMENT CALCULATION

Description

This program calculates the monthly payment necessary to fully amortize a loan. You must provide the amount of the loan, the annual interest rate, and the term of the loan in years.

Example

Don and Donna can purchase a house for $100,000. They plan to make a $25,000 down payment and finance the balance at 12% for 30 years. Find their monthly payment. Figures 4.2 and 4.3 show the screen display and the program listing for the Monthly Payment program.

```
MONTHLY PAYMENT

  AMOUNT BORROWED ($)    75000
  ANNUAL INT RATE (%)    12
  TERM OF LOAN (YRS)     30

MONTHLY PAYMENT IS       771.46

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 4.2: Screen Display: Monthly Payment Calculation

```
610  REM -----------------
620 N$ = "MONTHLY PAYMENT"
630  REM -----------------
640  GOSUB 7500: REM INITIALIZE
650 K = 1
660  GOSUB 370: REM PARAMETERS
670  GOSUB 270: REM CALC PMT
680  PRINT
690  PRINT "MONTHLY PAYMENT IS"; TAB( 25);
700 P =  INT (P * 100 + .5) / 100
710  PRINT P
720  REM  ASK FOR ANOTHER
730 Q2$ = "COMPUTE"
740  GOSUB 4500
750  IF YN$ = "N" THEN  END
760  GOTO 610
```

Figure 4.3: Program Listing: Monthly Payment Calculation

MORTGAGE SCHEDULE

Description

This program displays a mortgage schedule on the screen for any year you choose. You enter the amount of the loan, the annual interest rate, the term of the loan, and the year for which you wish the mortgage schedule to be displayed. Thus, you can rapidly review the principal and interest you pay in any year. The interest you pay is deductible from your income tax.

Example

The mortgage on Sam and Samantha's home is $60,000. The annual interest rate is 11.5%, and the term of the loan is 25 years. Sam and Samantha would like to review the principal and interest they will be paying during the sixth year of the loan. Figure 4.4 shows the screen display for this example. The program listing is displayed in Figure 4.5.

```
MORTGAGE SCHEDULE

  AMOUNT BORROWED ($)    60000
  ANNUAL INT RATE (%)    11.5
  TERM OF LOAN (YRS)     25
  YEAR                   6
MONTH   PRINCIPLE   INTEREST   BALANCE
61        61.82       548.06    57127.27
62        62.41       547.47    57064.86
63        63.01       546.87    57001.85
64        63.61       546.27    56938.24
65        64.22       545.66    56874.02
66        64.84       545.04    56809.18
67        65.46       544.42    56743.72
68        66.09       543.79    56677.63
69        66.72       543.16    56610.91
70        67.36       542.52    56543.55
71        68          541.88    56475.55
72        68.66       541.22    56406.89
---------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 4.4: Screen Display: Mortgage Schedule

```
770  REM ------------------
780 N$ = "MORTGAGE SCHEDULE"
790  REM -------------------
800 K = 1
810  GOSUB 7500: REM INITIALIZE
820  GOSUB 370: REM PARAMETERS
830  REM  GET YEAR
840 Q3$ = "YEAR"
850  GOSUB 8000:N = PAR(4)
860  GOSUB 270: REM CALC PMT
870  REM COMPUTE FIRST MONTH
880 N = 12 * N - 12
890  REM  GET BEGINNING BALANCE
900  GOSUB 490
910  PRINT "MONTH  PRINCIPLE  INTEREST  BALANCE"
920 P =  INT (100 * P + .5) / 100
930  FOR J = 1 TO 12
940  REM  COMPUTE P & I
950 I1 =  INT (IN / 12 * AM + .5) / 100:P1 = P - I1
960 AM = AM - P1:P1 =  INT (P1 * 100 + .5) / 100
970 AM =  INT (AM * 100 + .5) / 100
980  PRINT N + 1; TAB( 9);P1; TAB( 20);I1; TAB( 30);AM
990 N = N + 1
1000  NEXT J
1010 Q2$ = "COMPUTE": GOSUB 4500
1020  IF YN$ = "N" THEN  END
1030  GOTO 770
```

Figure 4.5: Program Listing: Mortgage Schedule

REMAINING BALANCE OF A LOAN

Description

This program finds the remaining balance on a loan. You can run this program to calculate the amount due should you choose to pay off the loan. The program assumes that all prior payments have been the exact monthly payment. To use the program you specify the amount of the mortgage, the interest rate and length of the loan, and the payment number for which you want to determine the remaining balance. Don't be concerned if this program takes a while to display its answer.

Example

John's mortgage is $90,000. His loan is for 30 years at an annual interest rate of 9%. He would like to compute the loan balance at the end of five years (the sixtieth payment). Figures 4.6 and 4.7 show the screen display and the program listing for the Remaining Balance program.

```
REMAINING BALANCE

  AMOUNT BORROWED ($)    90000
  ANNUAL INT RATE (%)    9
  TERM OF LOAN (YRS)     30
  PAYMENT NUMBER         60

MONTHLY PAYMENT         724.16
BAL AFTER 60 PMTS       86292.12

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 4.6: Screen Display: Remaining Balance of a Loan

```
1040  REM --------------------
1050 N$ = "REMAINING BALANCE"
1060  REM --------------------
1070 K = 1
1080  GOSUB 7500: REM INITIALIZE
1090  GOSUB 370: REM PARAMETERS
1100  REM  GET PAYMENT NUMBER
1110 Q3$ = "PAYMENT NUMBER"
1120  GOSUB 8000:N = PAR(4)
1130  GOSUB 270: REM CALC PMT
1140  GOSUB 490: REM COMPUTE BAL
1150  PRINT
1160  PRINT "MONTHLY PAYMENT"; TAB( 23);
1170 P =  INT (P * 100 + .5) / 100
1180  PRINT P
1190 AM =  INT (AM * 100 + .5) / 100
1200  PRINT "BAL AFTER ";N;" PMTS"; TAB( 23);AM
1210  REM  ASK FOR ANOTHER
1220 Q2$ = "COMPUTE": GOSUB 4500
1230  IF YN$ = "N" THEN  END
1240  GOTO 1040
```

Figure 4.7: Program Listing: Remaining Balance of a Loan

EFFECT OF ACCELERATED PAYMENTS

Description

You will find this program extremely useful—it allows you to find the effect of increasing the monthly payments on your mortgage. An increase in the monthly payments will reduce both the number of years of payments and the total interest you pay. To use this program, enter the amount of the loan, the annual interest rate, the length of the loan, the year in which you increase the payments, and the extra amount you will pay each month. This program computes as many times as necessary to reduce the loan balance to zero, so you may find that it takes a while for the answer to appear.

Example

Cindy has decided to increase the monthly payments on her home loan by $100 each month. She begins the increased payments in the seventh year of the loan and would like to know how much she will save on the interest. Her loan is for $60,000 at 9.75%; the term of her loan is 30 years. You can see in Figure 4.8 that Cindy will save $41,000 with this increase in her monthly payment. Figure 4.9 shows the program listing for the Accelerated Payments program.

```
1250  REM ----------------------
1260 N$ = "ACCELERATED PAYMENTS"
1270  REM ----------------------
1280 K = 1
1290  GOSUB 7500: REM INITIALIZE
1300  GOSUB 370: REM PARAMETERS
1310  REM  GET YEAR
1320 Q3$ = "YEAR OF INCREASE"
1330  GOSUB 8000:N = PAR(4)
1340  GOSUB 270: REM CALC PMT
1350  REM  COMPUTE FIRST MONTH
1360 N = 12 * N - 12
1370  REM  GET EXTRA
1380 Q3$ = "EXTRA AMOUNT PAID ($)"
1390  GOSUB 8000:EX = PAR(5)
1400  REM  COMP. PRIN INT & BAL
1410 TI = 0:BAL = 0:TP = 0
```

Figure 4.9: Program Listing: Effect of Accelerated Payments (continues)

```
ACCELERATED PAYMENTS

  AMOUNT BORROWED ($)    60000
  ANNUAL INT RATE (%)    9.75
  TERM OF LOAN (YRS)     30
  YEAR OF INCREASE       7
  EXTRA AMOUNT PAID ($)  100

YEARS TO PAY OFF         20.58
TOTAL INTEREST PAID      84432.16
INTEREST SAVED           41145.19

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 4.8: Screen Display: Effect of Accelerated Payments

```
1420  FOR J = 1 TO N
1430 I1 = IN / 12 / 100 * AM:P1 = P - I1:AM = AM - P1
1440 TI = TI + I1:TP = TP + P1
1450  NEXT J
1460  REM  COMPUTE INT W/ NO INC.
1470 T2 = 0:A2 = PAR(1)
1480  FOR J = 1 TO 12 * YR
1490 I2 = IN / 12 / 100 * A2:P2 = P - I2
1500 A2 = A2 - P2:T2 = T2 + I2
1510  NEXT J
1520  REM  COMPUTE WITH XTRA PMT
1530 J = 0
1540 I1 = IN / 12 / 100 * AM:TI = TI + I1
1550 P1 = P + EX - I1:TP = TP + P1
1560 AM = AM - P1:J = J + 1
1570  IF AM > 0 THEN 1540
1580  PRINT
1590  PRINT "YEARS TO PAY OFF"; TAB( 25);
1600 D = PAR(4) + J / 12 - 1
1610  PRINT  INT (D * 100 + .5) / 100
1620 TI =  INT (TI * 100 + .5) / 100
1630 T2 =  INT (T2 * 100 + .5) / 100
1640  PRINT "TOTAL INTEREST PAID"; TAB( 25);TI
1650  PRINT "INTEREST SAVED"; TAB( 25);T2 - TI
1660 Q2$ = "COMPUTE": GOSUB 4500
1670  IF YN$ = "N" THEN  END
1680  GOTO 1250
```

Figure 4.9: Program Listing: Effect of Accelerated Payments

BALLOON PAYMENT CALCULATION

Description

Many recent mortgages have been set up with a balloon payment due at the end of a specified period. This program allows you to calculate the required balloon payment. You must enter the amount of the loan, the annual interest rate, the term of the loan, and the year in which the balloon payment is due.

Example

Max has a mortgage for $125,000. The annual interest rate is 11%. The payments are amortized over 30 years; Max must pay off the entire loan at the end of 10 years. What will be the amount of Max's balloon payment? Figures 4.10 and 4.11 show the screen display and the program listing for the Balloon Payment program.

```
BALLOON PAYMENT

  AMOUNT BORROWED ($)    125000
  ANNUAL INT RATE (%)    11
  TERM OF LOAN (YRS)     30
  YEAR LOAN DUE          10

MONTHLY PAYMENT         1190.4

BALLOON PAYMENT
AFTER 10 YEARS          115328.2

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 4.10: Screen Display: Balloon Payment Calculation

```
1690  REM -----------------
1700 N$ = "BALLOON PAYMENT"
1710  REM -----------------
1720 K = 1
1730  GOSUB 7500: REM INITIALIZE
1740  GOSUB 370: REM PARAMETERS
1750  REM  GET YEAR LOAN DUE
1760 Q3$ = "YEAR LOAN DUE"
1770  GOSUB 8000:N = 12 * PAR(4)
1780  GOSUB 270: REM CALC PMT
1790  REM  COMPUTE BALANCE
1800  GOSUB 490
1810  PRINT
1820 P =  INT (P * 100 + .5) / 100
1830  PRINT "MONTHLY PAYMENT"; TAB( 23);P
1840 AM =  INT (AM * 100 + .5) / 100
1850  PRINT
1860  PRINT "BALLOON PAYMENT"
1870  PRINT "AFTER ";N / 12;" YEARS"; TAB( 23);AM
1880  REM  ASK FOR ANOTHER
1890 Q2$ = "COMPUTE"
1900  GOSUB 4500
1910  IF YN$ = "N" THEN  END
1920  GOTO 1690
```

Figure 4.11: Program Listing: Balloon Payment Calculation

AFFORDABLE HOUSE PRICE

Description

You can use this program to find the price of a home for which a bank will give you a loan. This program can be invaluable when you are negotiating to buy a new home. To use the program, enter the annual interest rate and term of available loans, your annual income, the estimated annual taxes and insurance, the percent of your gross income that the bank will allow for your total monthly payments, and the percent down payment you will make.

Example

Bill and Betty would like to purchase a new home. The available interest rate for home loans is 13.5% for 30-year loans. The couple's annual income is $50,000 per year, and they plan to put 20% down on their new house. Bill and Betty estimate that their yearly taxes and insurance will be $2,400. Their banker has told them that the monthly payment on their house must not exceed 35% of their monthly income. Figures 4.12 and 4.13 are the screen display and the program listing for the Affordable House Price program.

```
AFFORDABLE HOUSE PRICE

  ANNUAL INT RATE (%)    13.5
  TERM OF LOAN (YRS)     30
  BUYER'S ANNUAL INCOME  50000
  EST ANNUAL TAX & INS   2400
  % OF INC FOR PAYMENTS  35
  % DOWN PAYMENT         20

AMOUNT FINANCED          109858.56
AFFORDABLE HOUSE PRICE   137323.2




----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 4.12: Screen Display: Affordable House Price

```
1930  REM ------------------------
1940 N$ = "AFFORDABLE HOUSE PRICE"
1950  REM ------------------------
1960 K = 1
1970  GOSUB 7500: REM INITIALIZE
1980 Q1$ = "SPECIFY PARAMETERS:"
1990 Q2$ = ""
2000 Q3$ = "ANNUAL INT RATE (%)": GOSUB 8000
2010 Q3$ = "TERM OF LOAN (YRS)": GOSUB 8000
2020 Q3$ = "BUYER'S ANNUAL INCOME": GOSUB 8000
2030 Q3$ = "EST ANNUAL TAX & INS": GOSUB 8000
2040 Q3$ = "% OF INC FOR PAYMENTS": GOSUB 8000
2050 Q3$ = "% DOWN PAYMENT": GOSUB 8000
2060  PRINT
2070  REM  COMPUTE ALLOWABLE PAYMENT
2080 P = PAR(5) / 100 * PAR(3) / 12 - PAR(4) / 12
2090  REM  COMPUTE AMOUNT FINANCED
2100 IN = PAR(1) / 100 / 12:N1 = 12 * PAR(2)
2110 V = 1 / (1 + IN):AM = P * (1 - V ^ N1) / IN
2120 AM =  INT (AM * 100 + .5) / 100
2130  PRINT "AMOUNT FINANCED"; TAB( 25);AM
2140 D = AM / (1 - PAR(6) / 100)
2150 D =  INT (D * 100 + .5) / 100
2160  PRINT N$; TAB( 25);D
2170 Q2$ = "COMPUTE": GOSUB 4500
2180  IF YN$ = "N" THEN  END
2190  GOTO 1930
```

Figure 4.13: Program Listing: Affordable House Price

MORTGAGE WITH SECOND MORTGAGE

Description

We shall now examine a program you can use to find the total payments on a house that will have both a first and second mortgage. To use this program you enter the purchase price of the home, the cash available for a down payment, and the amount of the first mortgage you will obtain. You also specify the interest rate of the first and second mortgages and the length of the first. We assume that the second mortgage is paid monthly on an interest-only basis.

Example

Joanne has saved $20,000 for a down payment on a new house. The house will cost $140,000 and she plans to take out a first mortgage for $90,000. The first mortgage will be for 30 years at 12% annual interest. She will also have a 16% interest-only second mortgage. What will be her total payments? The resulting screen display for this example is shown in Figure 4.14; the program listing appears in Figure 4.15.

```
2200  REM ----------------------
2210 N$ = "MORTGAGE WITH SECOND"
2220  REM ----------------------
2230  GOSUB 7500:K = 1
2240 Q1$ = "SPECIFY PARAMETERS"
2250 Q2$ = ""
2260 Q3$ = "PURCHASE PRICE ($)": GOSUB 8000
2270 Q3$ = "CASH AVAILABLE ($)": GOSUB 8000
2280 Q3$ = "FIRST MORT ($)": GOSUB 8000
2290 Q3$ = "FIRST RATE (%)": GOSUB 8000
```

Figure 4.15: Program Listing: Mortgage with Second Mortgage (continues)

```
MORTGAGE WITH SECOND

  PURCHASE PRICE ($)      140000
  CASH AVAILABLE ($)      20000
  FIRST MORT ($)          90000
  FIRST RATE (%)          12
  FIRST TERM (YRS)        30
  SECOND RATE (%)         16

PAYMENT (FIRST)           925.75
PAYMENT (SECOND)          400
TOTAL PAYMENTS            1325.75




----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 4.14: Screen Display: Mortgage with Second Mortgage

```
2300 Q3$ = "FIRST TERM (YRS)": GOSUB 8000
2310 Q3$ = "SECOND RATE (%)": GOSUB 8000
2320  PRINT
2330 AM = PAR(3):IN = PAR(4):YR = PAR(5)
2340  REM  COMPUTE FIRST PMT
2350  GOSUB 270
2360 P =  INT (P * 100 + .5) / 100
2370  PRINT "PAYMENT (FIRST)"; TAB( 23);P
2380  REM  COMPUTE SECOND PMT
2390  IF PAR(2) < PAR(1) - PAR(3) THEN 2420
2400  PRINT "SECOND MORTGAGE NOT REQUIRED"
2410  GOTO 2470
2420 D = PAR(1) - PAR(2) - PAR(3)
2430 P2 = D * PAR(6) / 100 / 12
2440 P2 =  INT (P2 * 100 + .5) / 100
2450  PRINT "PAYMENT (SECOND)"; TAB( 23);P2
2460  PRINT "TOTAL PAYMENTS"; TAB( 23);P + P2
2470 Q2$ = "COMPUTE": GOSUB 4500
2480  IF YN$ = "N" THEN  END
2490  GOTO 2200
```

Figure 4.15: Program Listing: Mortgage with Second Mortgage

RENTAL PROPERTY ANALYSIS

Description

Let's now consider a program that is very useful for analyzing real estate investments. You can use the Rental Property Analysis program to find the monthly payment and cash flow on a rental property. (We don't consider the after-tax effects of depreciation.) You must enter the amount of the loan on your rental property, the annual interest rate and length of the loan, and the annual insurance, taxes, and maintenance. You must also enter the monthly rental income.

Example

Herb is planning to purchase a house, which he will rent for $475 per month. The first mortgage will be $80,000, amortized over 30 years at 12% interest. Herb estimates that the annual insurance will be $500, the property taxes $1,600, and the maintenance $400. What will be his monthly payment and cash flow? Figures 4.16 and 4.17 are the screen display and program listing for this example.

```
RENTAL PROPERTY ANALYSIS

  AMOUNT BORROWED ($)    80000
  ANNUAL INT RATE (%)    12
  TERM OF LOAN (YRS)     30
  ANNUAL INSURANCE ($)   500
  ANNUAL TAXES ($)       1600
  ANNUAL MAINT ($)       400
  MONTHLY INCOME ($)     475

MONTHLY PAYMENT IS     822.89
MONTHLY CASH FLOW      -556.22

----------------------------------------
WOULD YOU LIKE TO
COMPUTE
AGAIN? (Y OR N)
```

Figure 4.16: Screen Display: Rental Property Analysis

```
2500  REM --------------------------
2510 N$ = "RENTAL PROPERTY ANALYSIS"
2520  REM --------------------------
2530  GOSUB 7500:K = 1
2540  REM  GET PARAMETERS
2550  GOSUB 370
2560 Q3$ = "ANNUAL INSURANCE ($)": GOSUB 8000
2570 Q3$ = "ANNUAL TAXES ($)": GOSUB 8000
2580 Q3$ = "ANNUAL MAINT ($)": GOSUB 8000
2590 Q3$ = "MONTHLY INCOME ($)": GOSUB 8000
2600  GOSUB 270: REM CALC PMT.
2610  PRINT
2620 P =  INT (P * 100 + .5) / 100
2630  PRINT "MONTHLY PAYMENT IS"; TAB( 23);P
2640 CF = PAR(7) - P - (PAR(4) + PAR(5) + PAR(6)) / 12
2650 CF =  INT (CF * 100 + .5) / 100
2660  PRINT "MONTHLY CASH FLOW"; TAB( 23);CF
2670 Q2$ = "COMPUTE": GOSUB 4500
2680  IF YN$ = "N" THEN  END
2690  GOTO 2500
```

Figure 4.17: Program Listing: Rental Property Analysis

REAL ESTATE PROGRAMS MENU

We can again combine the programs in this chapter to provide a menu-driven real estate analysis package. Remember to replace the END command in each program with a RETURN command. Instructions for combining and saving the menu program are in Chapter 2. The menu display and its program listing are shown in Figures 4.18 and 4.19.

```
1= MONTHLY PAYMENT CALCULATION
2= MORTGAGE SCHEDULE
3= REMAINING BALANCE OF LOAN
4= EFFECT OF ACCELERATED PAYMENTS
5= BALLOON PAYMENT CALCULATION
6= AFFORDABLE HOUSE PRICE
7= MORTGAGE WITH SECOND
8= RENTAL PROPERTY ANALYSIS

----------------------------------------

CHOOSE PROGRAM:
```

Figure 4.18: Screen Display: Real Estate Programs Menu

```
100  REM ----------------------
110 N$ = "REAL ESTATE PROGRAMS"
120  REM ----------------------
130  GOSUB 7500: REM INITIALIZE
140 X$(1) = "MONTHLY PAYMENT CALCULATION"
150 X$(2) = "MORTGAGE SCHEDULE"
160 X$(3) = "REMAINING BALANCE OF LOAN"
170 X$(4) = "EFFECT OF ACCELERATED PAYMENTS"
180 X$(5) = "BALLOON PAYMENT CALCULATION"
190 X$(6) = "AFFORDABLE HOUSE PRICE"
200 X$(7) = "MORTGAGE WITH SECOND"
210 X$(8) = "RENTAL PROPERTY ANALYSIS"
220  REM  DISPLAY MENU
230 N = 8: GOSUB 8500
240  ON X GOSUB 610,770,1040,1250,1690,1930,2200,2500
250  GOTO 110
```

Figure 4.19: Program Listing: Real Estate Programs Menu

5 DATA ANALYSIS PROGRAMS

This chapter describes a set of data analysis programs that you can use to reduce and plot various kinds of data. We present programs that input data, plot it on the screen, compute moving averages, and perform linear regression analyses. The Plot Data program will also plot data that are reduced by the programs in this chapter.

Investors, among others, will find the data analysis programs very useful. You can input daily stock prices and then try the various smoothing programs. These programs smooth out daily price variations and give an indication of longer term trends. The data analysis programs can also be used in many other situations, including analyzing scientific data, plotting and smoothing weight variations, and tracking weather data.

The programs in this and the following two chapters require that you first enter the menu program and additional common subroutines. You should save these programs in one file after they are entered; you then add the other programs as described in Chapter 1. Because the programs in these chapters *begin* with the menu program, you don't have to change any lines within the programs to activate the menu.

DATA ANALYSIS PROGRAMS MENU

The data analysis programs all work together through common arrays and a common menu program. Notice that the menu lists an input program, a plot program, and several analysis programs. You use the programs together by first entering data with the input program. You can then plot the data or analyze it. Input data are stored in the array D. Data you reduce with the analysis programs are stored in the array R.

After you run each program you return to the menu. You can then analyze the data again with a different program; you can plot input data, reduced data, or both; or you can enter new data.

Figure 5.1 shows the screen display of the Data Analysis Menu. Figure 5.2 shows the menu program listing.

```
1= INPUT DATA
2= PLOT DATA
3= MEAN AND STANDARD DEVIATION
4= 3-PT MOVING AVERAGE
5= 3-PT WEIGHTED MOVING AVERAGE
6= 4-PT CENTERED AVERAGE
7= LINEAR REGRESSION

----------------------------------------

CHOOSE PROGRAM:
```

Figure 5.1: Screen Display: Data Analysis Menu

```
100  REM ------------------------
110 N$ = "DATA ANALYSIS PROGRAMS"
120  REM ------------------------
130  REM  SET UP ARRAYS
140  DIM P(40),R(40),D(40)
150  GOSUB 7500: REM INITIALIZE
160  REM  SET UP MENU
170 X$(1) = "INPUT DATA"
180 X$(2) = "PLOT DATA"
190 X$(3) = "MEAN AND STANDARD DEVIATION"
200 X$(4) = "3-PT MOVING AVERAGE"
210 X$(5) = "3-PT WEIGHTED MOVING AVERAGE"
220 X$(6) = "4-PT CENTERED AVERAGE"
230 X$(7) = "LINEAR REGRESSION"
240 N = 7
250  GOSUB 8500: REM DISPLAY MENU
260  ON X GOSUB 730,980,1250,1460,1680,1920,2150
270  GOTO 150
```

Figure 5.2: Program Listing: Data Analysis Menu

DATA ANALYSIS SUBROUTINES

The Plot Data program uses two subroutines to plot data on the screen. One routine draws the axes; the other scales and plots the data. You must enter these programs prior to using the plot program. Figure 5.3 is the program listing for the axis subroutine.

Figure 5.4 shows the program listing for the routine that scales and plots the data. This program plots either input data, reduced data, or both input and reduced data, depending on the user-controlled variable CH$.

```
280  REM ----------------------
290  REM         DRAW AXES
300  REM ----------------------
310  HOME
320  REM  Y AXIS
330  FOR I = 1 TO 21
340  VTAB I: HTAB 6: PRINT " I"
350  NEXT I
360  REM  Y AXIS SCALE
370  FOR I = 1 TO 16 STEP 5
380  VTAB I: HTAB 6: PRINT "-"
390  VTAB I: HTAB 1
400  PRINT MX - (MX - MN) / 4 * (I - 1) / 5
410  NEXT I
420  VTAB 21: HTAB 8
430  REM  X AXIS
440  FOR I = 1 TO 32
450  PRINT "-";
460  NEXT I
470  REM  X AXIS TICKS
480  FOR I = 8 TO 38 STEP 5
490  VTAB 21: HTAB I: PRINT "I";
500  VTAB 22: HTAB I: PRINT I - 7
510  NEXT I
520  RETURN
```

Figure 5.3: Program Listing: Axis Subroutine

```
530  REM ----------------------
540  REM  SCALE DATA & PLOT IT
550  REM ----------------------
560  FOR I = 1 TO ND
570 P1 = 21 -  INT (20 * (D(I) - MN) / (MX - MN))
580 P2 = 21 -  INT (20 * (R(I) - MN) / (MX - MN))
590  IF CH$ <  > "I" THEN 630
600  IF P1 > 21 OR P1 < 1 THEN 720
610  VTAB P1: HTAB I + 7: PRINT "*"
620  GOTO 710
630  IF CH$ <  > "R" THEN 670
640  IF P2 > 21 OR P2 < 1 THEN 720
650  VTAB P2: HTAB I + 7: PRINT "O"
660  GOTO 710
670  REM  MUST BE BOTH PLOTS
680  IF P1 > 21 OR P1 < 1 OR P2 > 21 OR P2 < 1 THEN 720
690  VTAB P1: HTAB I + 7: PRINT "*"
700  VTAB P2: HTAB I + 7: PRINT "O"
710  NEXT I: RETURN
720 ERF = 1: RETURN
```

Figure 5.4: Program Listing: Scale and Plot Subroutine

INPUT DATA

Description

You use this program to enter data that the other programs analyze and plot. You can enter up to 32 data points with the Input Data program. The data are stored in the array D(I). The program prompts you to enter data in the order of their occurrence. If you enter fewer than 32 data points, you type a D to signify that you are done entering data. The input screen is set up to display the last ten points that you enter, so you can check yourself as you go along. Remember, all data that you smooth and plot must first be entered with the Input Data program.

Example

Mary Ann would like to input the data she obtained in a laboratory experiment. The data are as follows:

Point	1	2	3	4	5	6	7	8	9	10
Data	1	3	5	4	4	5	7	8	9	11

Point	11	12	13	14	15	16	17	18	19	20
Data	12	15	13	14	15	16	15	13	12	12

Point	21	22	23	24	25	26	27	28	29	30
Data	11	10	7	9	8	7	5	5	4	3

Point	31	32
Data	3	0

Figure 5.5 shows the screen display; Figure 5.6 contains the program listing for the Input Data program.

```
INPUT DATA



   VALUE 10 = 11
   VALUE 11 = 12
   VALUE 12 = 15
   VALUE 13 = 13
   VALUE 14 = 14
   VALUE 15 = 15
   VALUE 6 = 5
   VALUE 7 = 7
   VALUE 8 = 8
   VALUE 9 = 9



----------------------------------------
TYPE IN DATA VALUES IN ORDER
TYPE D WHEN DONE
   VALUE 16 =?
```

Figure 5.5: Screen Display: Input Data

```
730  REM ------------
740 N$ = "INPUT DATA"
750  REM ------------
760  GOSUB 7500: REM INITIALIZE
770  REM  DIALOG
780 Q1$ = "TYPE IN DATA VALUES IN ORDER"
790 Q2$ = "TYPE D WHEN DONE"
800 Q3$ = "": GOSUB 6000
810 ND = 0: REM  NO. OF DATA PTS
820  FOR I = 1 TO 32
830  VTAB 23: HTAB 5
840  VTAB 23: HTAB 1: PRINT  SPC( 38): VTAB 23: HTAB 5
850  PRINT "VALUE ";I;" =";
860  INPUT D$
870  IF D$ = "D" THEN 970
880  VTAB I -  INT (I / 10) * 10 + 6
890  PRINT  SPC( 38)
900  VTAB I -  INT (I / 10) * 10 + 6: HTAB 1
910  PRINT "   VALUE ";I;" = ";D$
920  REM  COUNT DATA POINTS
930 ND = ND + 1
940  REM  STORE DATA
950 D(I) =  VAL (D$)
960  NEXT I
970  RETURN
```

Figure 5.6: Program Listing: Input Data

PLOT DATA

Description

The Plot Data program plots either input data or reduced data on the screen. (Reduced data result from using the data analysis programs that follow.) You must tell the program to plot input data, reduced data (data on which you have run one of the data analysis programs), or both by answering the program prompt with an I, R, or B respectively. The * symbol represents input data, and the 0 symbol represents reduced data. You must also specify the maximum and minimum Y-axis scale values. If your input data exceed your scale values, you will get the message DATA OUT OF RANGE. You must then respecify the scale values.

Example

Let's now plot the input data from the previous example. In this case we choose a maximum value of 20 and a minimum value of 0. We choose to plot input data (I), because we have not reduced it yet. Figure 5.7 shows the screen display for the data plot. Figure 5.8 displays the Plot Data program listing.

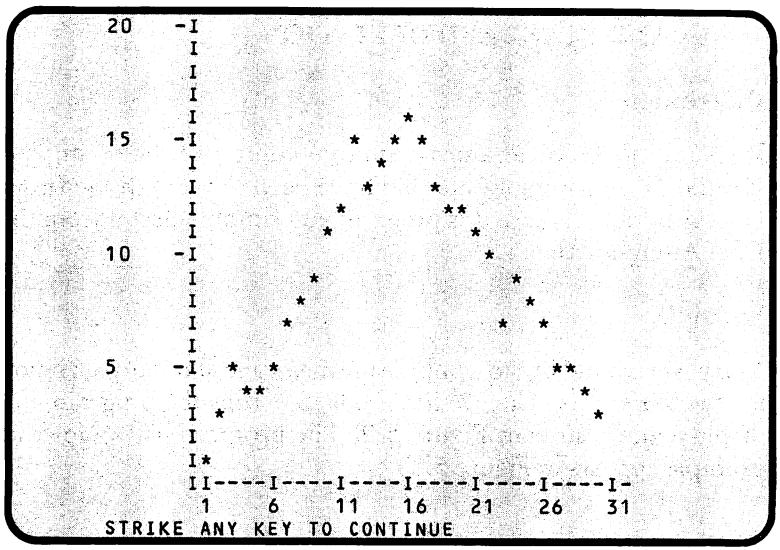


Figure 5.7: Screen Display: Plot Data

```
980  REM -------------------
990 N$ = "PLOT DATA ROUTINE"
1000  REM -------------------
1010 Q1$ = ""
1020 Q2$ = "INPUT OR REDUCED DATA"
1030 Q3$ = "OR BOTH (I OR R OR B)"
1040  GOSUB 6000
1050  GOSUB 6500: REM GET CHAR.
1060  IF CH$ = "I" OR CH$ = "R" OR CH$ = "B" THEN 1080
1070  GOTO 1050
1080 Q1$ = "":Q2$ = ""
1090 Q3$ = "CHOOSE MAX SCALE: "
1100  GOSUB 6000: INPUT MX
1110 Q1$ = "":Q2$ = ""
1120 Q3$ = "CHOOSE MIN SCALE: "
1130  GOSUB 6000: INPUT MN
1140  GOSUB 280: REM DRAW AXES
1150  GOSUB 530: REM PLOT DATA
1160  IF ERF <  > 1 THEN 1210
1170 ERF = 0
1180 Q1$ = "DATA OUT OF RANGE"
1190 Q2$ = "PLEASE RESPECIFY"
1200  GOTO 1090
1210  VTAB 24: HTAB 1
1220  PRINT "STRIKE ANY KEY TO CONTINUE";
1230  GET CH$
1240  RETURN
```

Figure 5.8: Program Listing: Plot Data

MEAN AND STANDARD DEVIATION

Description

We shall now look at a program that computes the mean and standard deviation for data that have been input with the Input Data program. To use this program you simply select it from the Data Analysis Menu.

Example

Mary Ann would like to compute the mean and standard deviation for the laboratory data she has input. She runs the program and displays the results in Figure 5.9. The program listing for this example appears in Figure 5.10.

```
MEAN AND STANDARD DEVIATION

MEAN OF INPUT DATA 8.76666667
STANDARD DEVIATION 4.23228334

---------------------------------------
STRIKE ANY KEY TO CONTINUE
```

Figure 5.9: Screen Display: Mean and Standard Deviation

```
1250  REM ----------------------------
1260 N$ = "MEAN AND STANDARD DEVIATION"
1270  REM ----------------------------
1280  GOSUB 7500
1290  REM  COMPUTE THE MEAN
1300 ME = 0
1310  FOR I = 1 TO ND
1320 ME = ME + D(I)
1330  NEXT I
1340 ME = ME / ND
1350  REM  COMPUTE STANDARD DEV.
1360 SD = 0
1370  FOR I = 1 TO ND
1380 SD = SD + (D(I) - ME) ^ 2
1390  NEXT I
1400 SD = (SD / ND) ^ .5
1410  PRINT
1420  PRINT "MEAN OF INPUT DATA"; TAB( 20);ME
1430  PRINT "STANDARD DEVIATION"; TAB( 20);SD
1440  GOSUB 9000
1450  RETURN
```

Figure 5.10: Program Listing: Mean and Standard Deviation

THREE-POINT MOVING AVERAGE

Description

We shall now present three data smoothing programs. The first program computes a three-point moving average on data you store in the Input Data array. The smoothed data are stored in the array R(I). You can display the smoothed data in either numeric or graph form. The three-point moving average takes the average value of the three prior data points and uses the average as the smoothed result. After the program smooths the data it will ask

```
DO YOU WANT TO PRINT THE DATA
```

If you answer with a Y, the program displays the data in numeric form. When you return to the main menu, you can graph the smoothed data with the Plot Data program. If you answer with an N, the program returns immediately to the main menu, so that you can run the Plot Data program.

Example

Mary Ann would like to try various smoothing techniques on her laboratory data. She first runs the Three-Point Moving Average program and graphs the results together with the input data. You can see the smoothed data in Figure 5.11; wherever the input data and smoothed data are the same, only the smoothed data are shown. The program listing is shown in Figure 5.12.

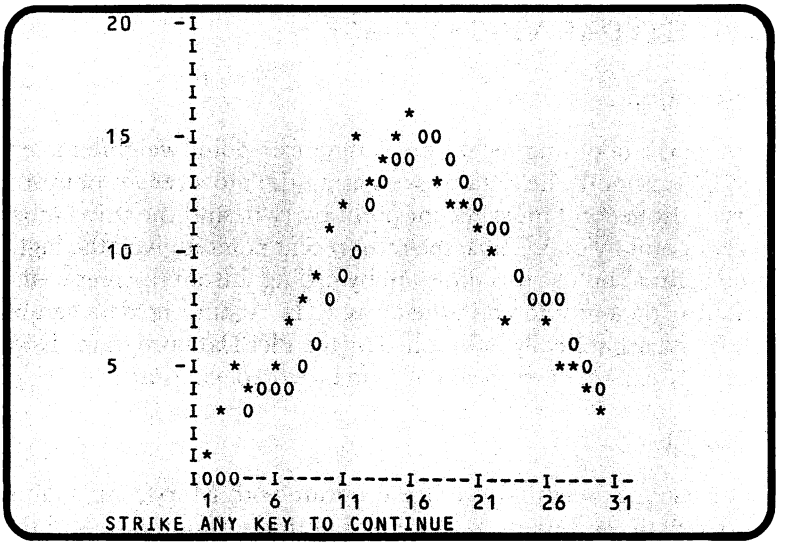


Figure 5.11: Screen Display: Three-Point Moving Average

```
1460  REM ---------------------
1470 N$ = "3-PT MOVING AVERAGE"
1480  REM ---------------------
1490  GOSUB 7500: REM INITIALIZE
1500  REM  SET UP RESULT IN R(I)
1510 R(1) = 0:R(2) = 0:R(3) = 0
1520  FOR I = 4 TO ND
1530 R(I) = (D(I - 3) + D(I - 2) + D(I - 1)) / 3
1540  NEXT I
1550 Q1$ = "DO YOU WANT TO PRINT THE"
1560 Q2$ = "DATA (Y OR N)"
1570 Q3$ = "": GOSUB 6000
1580  REM  ASK YES OR NO
1590  GOSUB 9500
1600  IF YN$ = "N" THEN  RETURN
1610  HOME
1620  FOR I = 4 TO ND
1630  PRINT R(I),
1640  NEXT I
1650  REM WAIT FOR KEY
1660  GOSUB 9000
1670  RETURN
```

Figure 5.12: Program Listing: Three-Point Moving Average

WEIGHTED MOVING AVERAGE

Description

A second smoothing technique is the three-point weighted average. We smooth the data by weighting the most recent point by three, the second most recent point by two, and the third most recent point by one. Thus, the most recent point is given the highest weight. You use this program by calling it from the menu; the resulting data are stored in the array R(I). Again, the data can be displayed numerically or graphed by the Plot Data program. Note that only one set of reduced data can be stored at a time.

Example

Let's use the weighted average program with Mary Ann's laboratory data. In Figure 5.13 we plot both the input data and the reduced data resulting from the Weighted Moving Average program. The program listing is given in Figure 5.14.

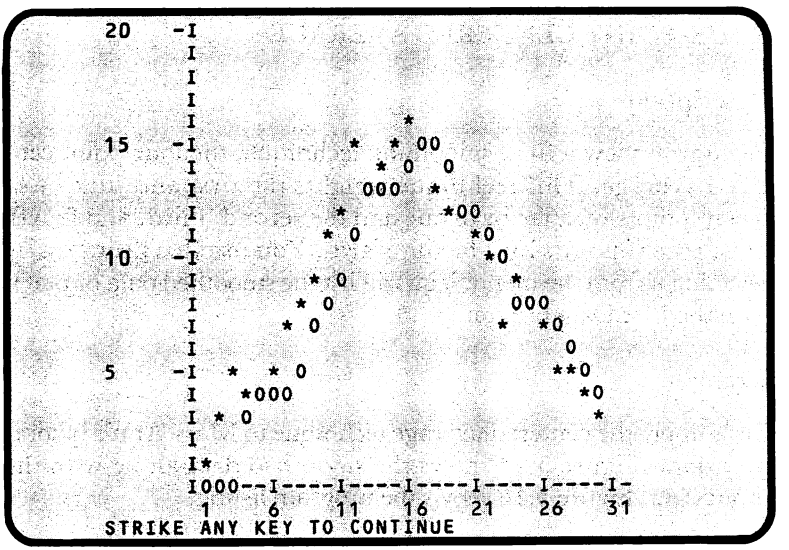


Figure 5.13: Screen Display: Weighted Moving Average

```
1680  REM ------------------------------
1690 N$ = "3-PT WEIGHTED MOVING AVERAGE"
1700  REM ------------------------------
1710  GOSUB 7500: REM INITIALIZE
1720  REM  SET UP RESULT IN R(I)
1730 R(1) = 0:R(2) = 0:R(3) = 0
1740  FOR I = 4 TO ND
1750 R(I) = (D(I - 3) + 2 * D(I - 2) + 3 * D(I - 1))
1760 R(I) = R(I) / 6
1770  NEXT I
1780 Q1$ = "DO YOU WANT TO PRINT THE"
1790 Q2$ = "DATA  (Y OR N)"
1800 Q3$ = ""
1810  GOSUB 6000
1820  REM  ASK YES OR NO
1830  GOSUB 9500
1840  IF YN$ = "N" THEN  RETURN
1850  HOME
1860  FOR I = 4 TO ND
1870  PRINT R(I),
1880  NEXT I
1890  REM WAIT FOR KEY
1900  GOSUB 9000
1910  RETURN
```

Figure 5.14: Program Listing: Weighted Moving Average

FOUR-POINT CENTERED AVERAGE

Description

We now review a final smoothing technique, the four-point centered average. This technique weights the first and fifth most recent points by a factor of one and the second, third, and fourth most recent points by a factor of two. You use this program by selecting it from the menu. You can list the smoothed data or plot it with the Plot Data program.

Example

Let's apply the centered-average technique to Mary Ann's laboratory data. Figure 5.15 shows the smoothed data along with the input data. Figure 5.16 shows the program listing.

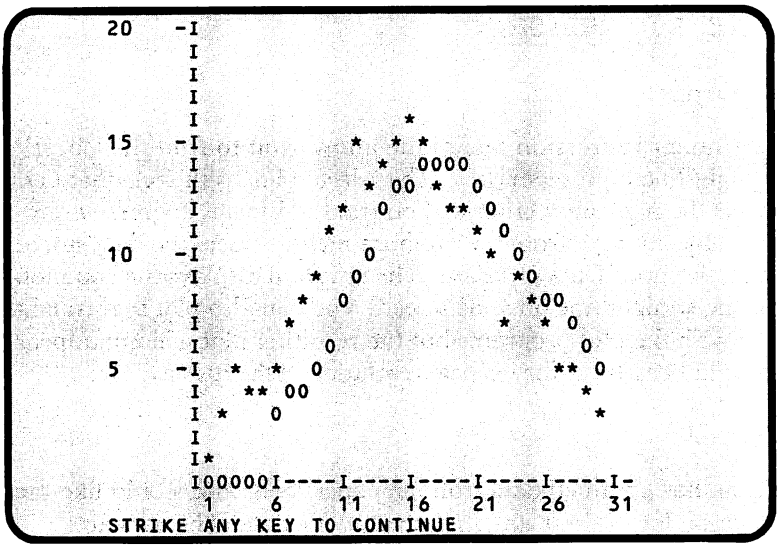


Figure 5.15: Screen Display: Four-Point Centered Average

```
1920  REM ----------------------
1930 N$ = "4-PT CENTERED AVERAGE"
1940  REM ----------------------
1950  GOSUB 7500: REM INITIALIZE
1960  REM  SET UP RESULT IN R(I)
1970 R(1) = 0:R(2) = 0:R(3) = 0:R(4) = 0:R(5) = 0
1980  FOR I = 6 TO ND
1990 D = D(I - 4) + D(I - 3) + D(I - 2)
2000 R(I) = (D(I - 5) + 2 * D + D(I - 1)) / 8
2010  NEXT I
2020 Q1$ = "DO YOU WANT TO PRINT THE"
2030 Q2$ = "DATA  (Y OR N)"
2040 Q3$ = "": GOSUB 6000
2050  REM  ASK YES OR NO
2060  GOSUB 9500
2070  IF YN$ = "N" THEN  RETURN
2080  HOME
2090  FOR I = 4 TO ND
2100  PRINT R(I),
2110  NEXT I
2120  REM WAIT FOR KEY
2130  GOSUB 9000
2140  RETURN
```

Figure 5.16: Program Listing: Four-Point Centered Average

LINEAR REGRESSION

Description

The linear regression technique allows you to find the "best" straight line for a set of data. The selected line is the one that best meets the regression criteria. You run the Linear Regression program by calling it from the program menu. It uses the data stored with the Input Data program. The program displays the equation for the straight line on your screen. You can also plot the straight line with the plot program, but the resulting plot will not appear straight because of the vertical resolution of the display.

Example

Frank has obtained data from an experiment and would like the equation for the best straight line for these data. The data are:

Point	1	2	3	4	5	6	7	8	9	10	11
Data	1	2	3	5	5	6	7	9	10	11	11

Point	12	13	14	15	16	17	18	19	20	21	22
Data	12	13	14	14	15	16	17	19	20	21	22

The screen display for this example appears in Figure 5.17. The program listing is shown in Figure 5.18.

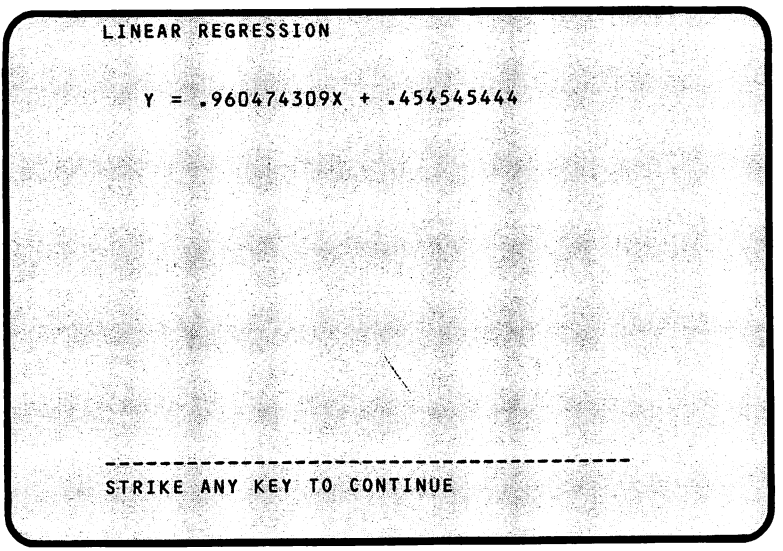


Figure 5.17: Screen Display: Linear Regression

```
2150  REM -------------------
2160 N$ = "LINEAR REGRESSION"
2170  REM -------------------
2180  GOSUB 7500: REM INITIALIZE
2190 SX = 0:SY = 0:SZ = 0:S2 = 0
2200  FOR I = 1 TO ND
2210 SX = SX + I
2220 SY = SY + D(I)
2230 XY = XY + D(I) * I
2240 S2 = S2 + I ^ 2
2250  NEXT I
2260 M = (XY - SX * SY / ND) / (S2 - SX ^ 2 / ND)
2270 B = SY / ND - M * SX / ND
2280  PRINT
2290  PRINT "    Y = ";M;"X + ";B
2300  REM  UPDATE RESULT ARRAY
2310  FOR I = 1 TO ND
2320 R(I) = B + M * I
2330  NEXT I
2340  GOSUB 9000
2350  RETURN
```

Figure 5.18: Program Listing: Linear Regression

6

RECORD KEEPING PROGRAMS

This chapter shows you how to use the diskette storage system to store, retrieve, and analyze data. We shall first create a personal file that you can use to store names and addresses. You can use the personal files you create for phone directories, business mailing lists, and other applications. Then we'll show you how to create an automobile file that will make it easy to keep mileage records for your car.

Disk file programs usually don't appear in collections of programs, because they can be very complicated. But by using our central subroutine library, we have been able to reduce the usual complexity of these disk file programs. The results are programs that are easy to enter and use.

Prior to using the programs in this chapter, you must add several new subroutines to the SUBLIB file. These subroutines begin at line numbers 4000, 5000, and 5500; the subroutine listings are in Appendix A.

Unlike the other chapters, the programs in this chapter must be saved in *two* files—one containing the Personal File programs and the other containing the Automobile File programs. Each set of programs begins with a menu.

These programs automatically create data files on disk drive A. You can add these files to your program disk, or you can use a freshly formatted disk.

First let's define a few terms. An *item* is the smallest quantity stored in a file. For the personal file the items are last name, first name, telephone number, street address, city, state, and zip code. The items in the auto record are date, odometer reading, and gallons of gas used. A *record* is defined as a set of items. For the personal file a record is the name, address, and phone number for each individual. Finally, a *file* is a set of records.

PERSONAL FILE MENU

The programs for the personal file run from a menu. As you can see, the individual programs allow you to create a file, add records, delete records, list the entire file, and search for individual records in the file. When you begin, you will create your file using the Create File program. You then add records to your file with the Add Record program. The Delete Record program removes records from your file; you also change a record by deleting it and then reentering it with new information. The List File program displays the entire file on your screen. The Search Record program allows you to find all records with the same last name. Figure 6.1 shows the menu screen display. Figure 6.2 shows the menu program listing.

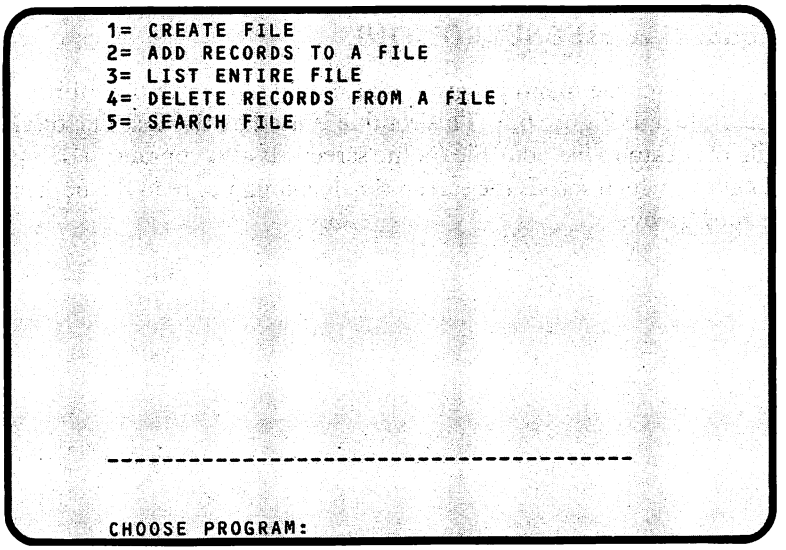


Figure 6.1: Screen Display: Personal File Menu

```
100  REM ---------------
110 N$ = "PERSONAL FILE"
120  REM ---------------
127  PRINT  SPC( 255) SPC( 185): VTAB 14
130  DIM S$(10,8)
140  GOSUB 7500: REM INITIALIZE
150 X$(1) = "CREATE FILE"
160 X$(2) = "ADD RECORDS TO A FILE"
170 X$(3) = "LIST ENTIRE FILE"
180 X$(4) = "DELETE RECORDS FROM A FILE"
190 X$(5) = "SEARCH FILE"
200  REM  DECLARE FIELDS
210 F$(1) = "LAST NAME"
220 F$(2) = "FIRST NAME"
230 F$(3) = "TELEPHONE"
240 F$(4) = "STREET ADDRESS"
250 F$(5) = "CITY"
260 F$(6) = "STATE"
270 F$(7) = "ZIP"
280 NF = 7:F$ = "PERSONAL FILE.TEXT"
290 FB$ = "PERSONAL FILE.BACKUP"
300 N = 5: GOSUB 8500: REM  MENU
310  ON X GOSUB 430,730,790,980,1080
320  GOTO 140
```

Figure 6.2: Program Listing: Personal File Menu

PERSONAL FILE SUBROUTINE

As you enter the menu program you should also enter the subroutine shown in Figure 6.3. This routine is used by the List Personal File program to list your file on the screen. It also contains a delay function, which scrolls the screen slowly enough to read the names as they go by.

```
330  REM ----------------------
340  REM  LIST RECORD ON SCREEN
350  REM ----------------------
360  PRINT L;".",P1$;", ";P2$
370  PRINT ,P3$
380  PRINT ,P4$
390  PRINT ,P5$;", ";P6$;" ";P7$
400  PRINT
410  FOR I1 = 1 TO 500: NEXT I1
420  RETURN
```

Figure 6.3: Program Listing: List Personal File Subroutine

CREATE FILE AND ADD RECORD

Description

The Create File program is used when you first set up your personal file. This program initializes the disk and disk directory and prepares the disk for data entry. If you use Create File on a disk already containing a personal file, the previous file will be erased. (The program warns you that the file is going to be erased, and it asks your permission to proceed.) Select the Create File program by displaying the Personal File Menu and pressing key 1.

The Add Record program is the one you use to add information to your file. You run the program by displaying the Personal File Menu and depressing the 2 on your keyboard. As with the Create File program, you are asked to enter the information that will appear in your file: last name, first name, telephone number, street address, city, state, and zip code. Records are stored in the order in which they are added.

Example

Figure 6.4 shows a screen display for a typical Add Record dialog. The Create File program had to be used first to create the file. Figure 6.5 shows the listings for both the Create File and Add Record programs.

```
430  REM ------------
440 N$ = "CREATE FILE"
450  REM ------------
460  GOSUB 7500: REM INITIALIZE
470 Q1$ = "CURRENT FILE WILL BE DELETED"
480 Q2$ = "":Q3$ = "PROCEED? (Y OR N)"
490  GOSUB 6000: GOSUB 9500
500  IF YN$ = "N" THEN  RETURN
510 K = 1
520 Q1$ = "ENTER DATA FOR THE FIRST NAME"
530 Q2$ = ""
```

Figure 6.5: Program Listing: Create File and Add Record (continues)

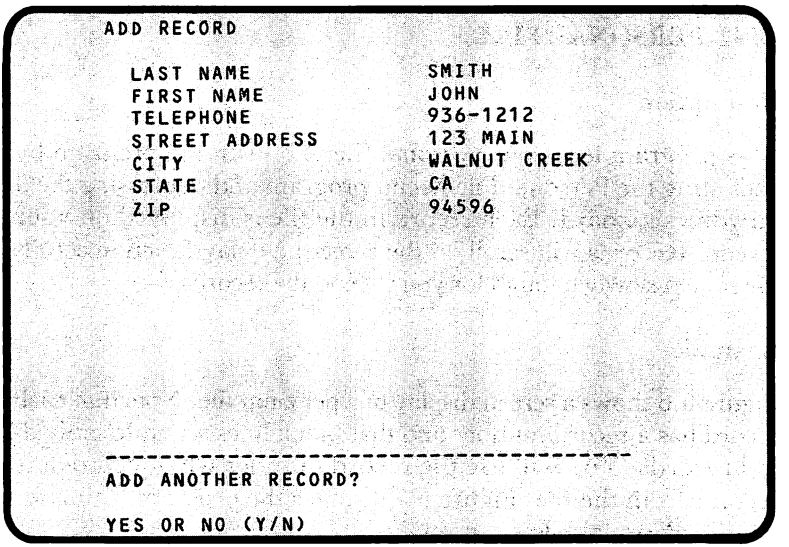


Figure 6.4: Screen Display: Add Record

```
540  FOR L = 1 TO 7
550 Q3$ = F$(L): GOSUB 7000
560  NEXT L
570  PRINT  CHR$ (4);"OPEN";F$
580  PRINT  CHR$ (4);"DELETE";F$
590  PRINT  CHR$ (4);"OPEN";F$
600  PRINT  CHR$ (4);"WRITE";F$
610 N = 2: PRINT N: FOR L = 1 TO 8: PRINT : NEXT
620  FOR L = 1 TO 8
630  PRINT P$(L)
640  NEXT L
650  PRINT  CHR$ (4);"CLOSE"
660  GOSUB 7500
670  VTAB 5: HTAB 1
680  PRINT "THE NEW FILE IS STARTED"
690  PRINT
700  PRINT "NOW USE ADD RECORD TO ADD TO IT"
710  GOSUB 9000: REM PAUSE
720  RETURN
730  REM ------------
740 N$ = "ADD RECORD"
750  REM ------------
760  REM  CALL ADD REC. SUB.
770  GOSUB 4000
780  RETURN
```

Figure 6.5: Program Listing: Create File and Add Record

LIST PERSONAL FILE

Description

This program lists your personal file. You run the program by activating the Personal File Menu program and depressing the 3 on your keyboard. Each record in the file is displayed on your screen. Records will scroll on the screen; a delay function scrolls the screen slowly enough for you to read the records.

Example

Figure 6.6 shows a screen display of a personal file. Note that each record has a record number and that blank lines separate individual records. You will use the record number when you delete records from the file. Figure 6.7 displays the program listing for the List Personal File program.

```
1.              SMITH, JOHN
                936-1212
                123 MAIN
                WALNUT CREEK, CA 94596

2.              SMITH, LISA
                936-1212
                123 MAIN
                WALNUT CREEK, CA  94596

3.              O'BRIEN, RONALD
                767-9080
                43 PANORAMIC
                BERKELEY, CA 94710

----------------------------------------
STRIKE ANY KEY TO CONTINUE
```

Figure 6.6: Screen Display: List Personal File

```
790  REM -----------
800 N$ = "LIST FILE"
810  REM -----------
820  GOSUB 7500: REM INITIALIZE
830  HOME : VTAB 5:L = 0
840  PRINT  CHR$ (4);"OPEN";F$
850  PRINT  CHR$ (4);"READ";F$
860  INPUT N: FOR L = 1 TO 8: INPUT P$: NEXT
865  IF N = 1 THEN 940
870 L = 0: FOR I = 1 TO N - 1
880 L = L + 1
890  PRINT  CHR$ (4);"READ";F$
900  INPUT P1$,P2$,P3$,P4$,P5$,P6$,P7$,P8$
910  PRINT  CHR$ (4)
920  GOSUB 330: REM DISPLAY NAME
930  NEXT I
940  PRINT  CHR$ (4);"CLOSE"
950  PRINT : PRINT : PRINT : PRINT
960  GOSUB 9000: REM PAUSE
970  RETURN
```

Figure 6.7: Program Listing: List Personal File

DELETE RECORD

Description

This program is used to delete old information from your personal file. You can correct information by first deleting it with the Delete Record program and then reentering correct data with the Add Record program. You use this program by indicating the number of the record you wish to delete. All subsequent records are renumbered so that the numbering remains consecutive. You can obtain the record number from the List Personal File program or the Search Personal File program. Figure 6.8 is the Delete Record program listing.

```
980  REM ---------------
990 N$ = "DELETE RECORD"
1000  REM ---------------
1010  GOSUB 5500: REM DELETE REC.
1020  IF L <  >  - 1 THEN 1040
1030  VTAB 18: HTAB 1: PRINT "RECORD NOT FOUND"
1040  REM  ASK FOR ANOTHER
1050 Q2$ = "DELETE": GOSUB 4500
1060  IF YN$ = "N" THEN  RETURN
1070  GOTO 1010
```

Figure 6.8: Program Listing: Delete Record

SEARCH PERSONAL FILE

Description

The Search Personal File program allows you to find an individual record without listing the entire file. You enter the last name of the individual you are searching for. The first record having the same last name you specify is displayed on your screen. The program then asks if you would like to search for additional matches. The program tells you if no last name matches are found.

The spelling you specify must match exactly the spelling in your file. Therefore, you must be consistent in the way you add records to your file. If you enter DU BOIS as the last name in your record and you search for the name DUBOIS, the program will tell you that no match is found.

Example

Let's search for the record for RONALD O'BRIEN. The screen display is shown in Figure 6.9. Figure 6.10 is the Search Personal File program listing.

```
1080  REM -------------
1090 N$ = "SEARCH FILE"
1100  REM -------------
1110  GOSUB 7500:F = 0:L =  - 1
1120 Q1$ = "SEARCH FOR LAST NAME"
1130 Q2$ = "":Q3$ = "LAST NAME: "
1140  GOSUB 6000
1150  INPUT NA$
1160  HOME
1170  VTAB 5: HTAB 1
1180  PRINT  CHR$ (4);"OPEN";F$
```

Figure 6.10: Program Listing: Search Personal File (continues)

```
3.              O'BRIEN, RONALD
                767-9080
                43 PANORAMIC
                BERKELEY, CA 94710




----------------------------------------
WOULD YOU LIKE TO SEARCH
FOR ADDITIONAL MATCHES (Y OR N)
```

Figure 6.9: Screen Display: Search Personal File

```
1190  PRINT  CHR$ (4);"READ";F$
1200  INPUT N: FOR A = 1 TO N
1210  PRINT  CHR$ (4);"READ";F$
1220 L = L + 1
1230  INPUT P1$,P2$,P3$,P4$,P5$,P6$,P7$,P8$
1240  IF P1$ <  > NA$ THEN 1310
1250  REM  DISPLAY RECORD
1260  PRINT  CHR$ (4):F = 1: VTAB 14
1270  PRINT  SPC( 255) SPC( 145): VTAB 14
1275  GOSUB 330: VTAB 24: PRINT  SPC( 200)
1280 Q1$ = "WOULD YOU LIKE TO SEARCH":Q3$ = ""
1290 Q2$ = "FOR ADDITIONAL MATCHES (Y OR N)"
1300  GOSUB 6000: GOSUB 9500: IF YN$ = "N" THEN 1320
1310  NEXT A: REM OUT OF RECORDS
1320  PRINT  CHR$ (4);"CLOSE"
1340  VTAB 18: HTAB 1
1350  IF F = 0 THEN  PRINT "NO MATCHES FOUND"
1360  REM  ASK FOR ANOTHER
1370 Q2$ = "SEARCH": GOSUB 4500
1380  IF YN$ = "N" THEN  RETURN
1390  GOTO 1080
```

Figure 6.10: Program Listing: Search Personal File

AUTOMOBILE FILE MENU

We will now develop a disk file that keeps mileage records for your automobile. You can use these programs to enter a calendar date, odometer reading, and amount of gasoline purchased. The List Automobile File program computes the average miles per gallon and returns that figure in the listing. You call the Automobile Record programs from a menu program. Figure 6.11 shows the screen display for the Automobile Record menu. Figure 6.12 is the menu program listing.

```
1= CREATE FILE
2= ADD RECORD
3= LIST FILE
4= DELETE RECORD

----------------------------------------

CHOOSE PROGRAM:
```

Figure 6.11: Screen Display: Automobile Record Menu

```
100  REM -----------------
110 N$ = "AUTOMOBILE FILE"
120  REM -----------------
130  GOSUB 7500: REM INITIALIZE
140 X$(1) = "CREATE FILE"
150 X$(2) = "ADD RECORD"
160 X$(3) = "LIST FILE"
170 X$(4) = "DELETE RECORD"
180  REM  DECLARE FIELDS
190 F$(1) = "DATE"
200 F$(2) = "ODOMETER"
210 F$(3) = "GALLONS"
220 NF = 3:F$ = "CAR.TEXT"
230 FB$ = "CAR.BACKUP"
240 N = 4: GOSUB 8500: REM MENU
250  ON X GOSUB 270,540,600,870
260  GOTO 130
```

Figure 6.12: Program Listing: Automobile Record Menu

CREATE FILE AND ADD RECORD

Description

The Create File program is used when you first set up your automobile file. This program initializes the disk and disk directory and prepares the disk for data entry. Run this program only once. If you want to keep records on more than one car, you must use different diskettes.

You use the Add Record program to add records to your automobile file. You follow the format established in the Create File program by entering the date, odometer reading, and gallons of gasoline purchased. Again, records are stored in the order in which they are added. If you make an error as you create a record, the average miles per gallon may be incorrectly computed. You should check your records carefully immediately after you enter them, because an incorrect record is difficult to correct once you have added other records.

Example

On February 17, 1983, you purchase 14.9 gallons of gas with an odometer reading of 13,523 miles. Figure 6.13 shows the screen display as you add this information to the file you previously created. Figure 6.14 is the program listing for the Create File and Add Record programs.

```
270  REM -------------
280 N$ = "CREATE FILE"
290  REM -------------
300  GOSUB 7500: REM INITIALIZE
310 K = 1
320 Q1$ = "CURRENT FILE WILL BE DELETED"
330 Q2$ = "":Q3$ = "PROCEED? (Y OR N)"
340  GOSUB 6000: GOSUB 9500: IF YN$ = "N" THEN  RETURN
350 Q1$ = "ENTER DATA FOR FIRST ENTRY"
360 Q2$ = ""
370 Q3$ = "DATE": GOSUB 7000
```

Figure 6.14: Program Listing: Create File and Add Record (continues)

```
ADD RECORD

  DATE                    2/17/83
  ODOMETER                13523
  GALLONS                 14.9

  ----------------------------------------
ADD ANOTHER RECORD?

YES OR NO (Y/N)
```

Figure 6.13: Screen Display: Add Record

```
380 Q3$ = "ODOMETER": GOSUB 7000
390 Q3$ = "GALLONS": GOSUB 7000
400  PRINT  CHR$ (4);"OPEN";F$
410  PRINT  CHR$ (4);"DELETE";F$
420  PRINT  CHR$ (4);"OPEN";F$
430  PRINT  CHR$ (4);"WRITE";F$
440 N = 2: PRINT N
450  FOR L = 1 TO 8: PRINT : NEXT : FOR I = 1 TO 8
460  PRINT P$(I): NEXT I
470  PRINT  CHR$ (4);"CLOSE";F$
480  GOSUB 7500: VTAB 5: HTAB 1
490  PRINT "THE NEW FILE IS STARTED"
500  PRINT
510  PRINT "NOW USE ADD RECORD TO ADD TO IT"
520  GOSUB 9000: REM PAUSE
530  RETURN
540  REM ------------
550 N$ = "ADD RECORD"
560  REM ------------
570  REM  CALL ADD RECORD SUB.
580  GOSUB 4000
590  RETURN
```

Figure 6.14: Program Listing: Create File and Add Record

LIST AUTOMOBILE FILE

Description

The List Automobile File program lists the contents of your automobile record on your screen. This program also computes and displays the average miles per gallon for each record. The screen will scroll slowly enough for you to read the information.

You can easily monitor your car's performance with this program. A sudden drop in average miles per gallon may indicate a problem with your car.

Example

You would like to view the contents of your automobile log. Figure 6.15 is the screen display for this example. Figure 6.16 is the List Automobile File program listing.

```
600  REM ----------
610 N$ = "LIST FILE"
620  REM ----------
630  GOSUB 7500: REM INITIALIZE
640 MI = 0:GAL = 0
650  PRINT "NO. DATE"; TAB( 15);"MILES";
660  PRINT  TAB( 23);"GAL"; TAB( 30);"AVERAGE"
670  PRINT  CHR$ (4);"OPEN";F$
680  PRINT  CHR$ (4);"READ";F$
```

Figure 6.16: Program Listing: List Automobile File (continues)

```
LIST FILE

NO.  DATE       MILES   GAL    AVERAGE
1    1/31/83    12730   15.6
2    2/7/83     12925   13.1   14.89
3    2/12/83    13210   15.8   18.04
4    2/17/83    13523   14.9   21.01
5    2/22/83    13801   15.2   18.29

----------------------------------------
STRIKE ANY KEY TO CONTINUE
```

Figure 6.15: Screen Display: List Automobile File

```
690  INPUT N: FOR L = 1 TO 8: INPUT P$: NEXT
695  IF N = 1 THEN 830
700  FOR I = 1 TO N - 1
710  PRINT  CHR$ (4);"READ";F$
720  INPUT P1$,P2$,P3$,P4$,P5$,P6$,P7$,P8$
730  PRINT  CHR$ (4)
740 GAL =  VAL (P3$)
750 AVE = ( VAL (P2$) - MI) / GAL
760 MI =  VAL (P2$)
770  PRINT I; TAB( 5);P1$; TAB( 15);P2$; TAB( 23);P3$;
780  IF I = 1 THEN  PRINT
785  LET AVE =  INT (AVE * 100 + .5) / 100
790  IF I <  > 1 THEN  PRINT  TAB( 30);AVE
800  REM DELAY TO READ ON SCREEN
810  FOR I1 = 1 TO 500
820  NEXT I1,I
830  PRINT  CHR$ (4);"CLOSE"
840  PRINT : PRINT : PRINT : PRINT
850  GOSUB 9000: REM PAUSE
860  RETURN
```

Figure 6.16: Program Listing: List Automobile File

DELETE RECORD

Description

The Delete Record program allows you to delete a record from your file. You can use this program to delete information that you enter incorrectly. You delete a record by specifying its number. When you delete a record in the middle of a file, all subsequent records are renumbered so that there are no holes in the file. Figure 6.17 shows the Delete Record program listing. You must delete records immediately upon incorrect entry; otherwise, the program will incorrectly compute miles per gallon.

```
870  REM ---------------
880 N$ = "DELETE RECORD"
890  REM ---------------
900  REM  CALL DELETE RECORD
910  GOSUB 5500
920  IF L <  >  - 1 THEN 940
930  VTAB 18: PRINT "RECORD NOT FOUND"
940  REM ASK FOR ANOTHER
950 Q2$ = "DELETE": GOSUB 4500
960  IF YN$ = "N" THEN  RETURN
970  GOTO 910
```

Figure 6.17: Program Listing: Delete Record

7

MATHEMATICS PRACTICE PROGRAMS

This chapter presents three sets of programs that allow you to use your computer to practice arithmetic. You must save each set in its own file. Each set runs from a menu, which you enter first. We provide drills in addition, subtraction, multiplication, and division, with both whole numbers and fractions. The three menus provide programs at multiple levels of difficulty; the programs are suitable for students from elementary to high school level. Each program keeps track of right and wrong answers. You will be greeted with a beep when you answer incorrectly. You will find that these programs can greatly sharpen your math skills; they are also helpful in completing homework assignments!

MATHEMATICS PRACTICE I MENU

The first set of four programs provides practice in adding, subtracting, multiplying, and dividing two whole numbers. You begin this set of programs by displaying the Mathematics Practice I Menu. Depress a number key to choose the drill you want. Once you have chosen the drill, the program asks you to choose the largest number you want to work with. Then the program presents you with two numbers and an operator to indicate addition, subtraction, multiplication, or division. You enter your answer from left to right. The programs generate a beep when you answer incorrectly. Each program keeps track of the total number of correct and incorrect answers. (Be sure to challenge yourself by choosing a large number!)

To use these programs you first enter the menu program and two common subroutines. Figure 7.1 shows the screen display for the menu program. Figure 7.2 is the program listing of the menu program.

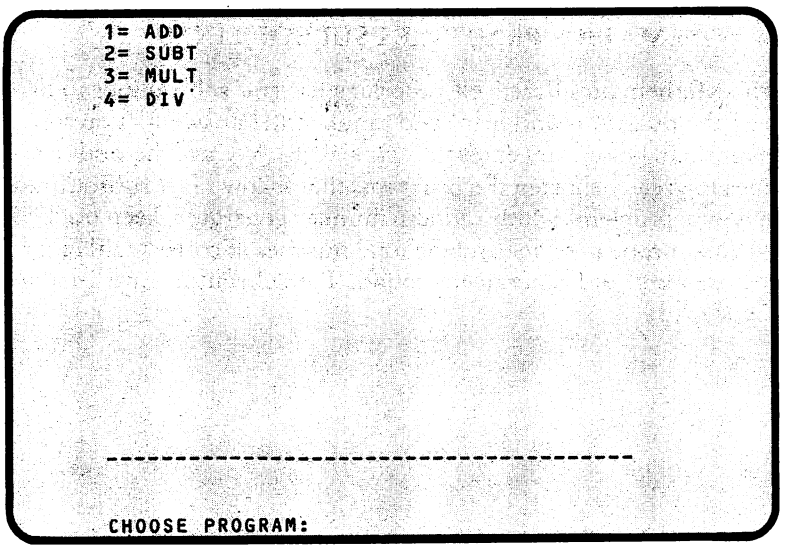


Figure 7.1: Screen Display: Mathematics Practice I Menu

```
100  REM ------------------------
110 N$ = "MATHEMATICS PRACTICE I"
120  REM ------------------------
130  GOSUB 7500: REM INITIALIZE
140  FOR X = 1 TO 1000: NEXT
150 K = 1
160  REM RUN MENU AND CHOOSE
170  REM TYPE OF DRILL
180 X$(1) = "ADD":X$(2) = "SUBT"
190 X$(3) = "MULT":X$(4) = "DIV"
200 N = 4
210  GOSUB 8500
220 R = 0:W = 0
230 Q1$ = "WHAT IS THE LARGEST NUMBER"
240 Q2$ = "YOU WOULD LIKE TO"
250 Y = X
260 Q3$ = X$(Y)
270  REM  ASK QUESTIONS
280  GOSUB 6000
290  PRINT  TAB( 20);: INPUT X
300  ON Y GOSUB 590,760,970,1150
310  GOTO 100
```

Figure 7.2: Program Listing: Mathematics Practice I Menu

MATHEMATICS PRACTICE I SUBROUTINES

The Mathematics Practice I programs use two subroutines to display the questions and print and tabulate the answers. You must enter them before you enter the actual drills. We use these subroutines to greatly shorten the programs that follow. The subroutines generate problems with a random number generator, keep track of the total problems done and the total number of correct and incorrect answers, and generate the sound. The subroutines are listed in Figure 7.3.

```
320  REM ----------------------
330  REM  DISPLAY QUESTION AND
340  REM  GET ANSWER
350  REM ----------------------
360 S1 = 4 -  LEN ( STR$ (N1))
370  VTAB 5: HTAB 20: PRINT N1$; SPC( S1);N1
380 S2 = 4 -  LEN ( STR$ (N2))
390  VTAB 6: HTAB 19: PRINT SN$; SPC( S2);N2
400  VTAB 7: HTAB 20: PRINT "----"
410  VTAB 10: PRINT "TYPE ANSWER:      ";
420  INPUT A
430  VTAB 8
440  PRINT "CORRECT ANSWER IS: ";
450  RETURN
460  REM ----------------------
470  REM   FINISH THE PROBLEM
480  REM ----------------------
490  VTAB 12: PRINT "YOU ARE ";
500  REM  UPDATE SCORE
510  IF S = 1 THEN R = R + 1: PRINT "RIGHT!"
520  IF S <  > 1 THEN W = W + 1: PRINT "WRONG."; CHR$ (7)
530  PRINT "PROBLEMS DONE:"; TAB( 20);R + W
540  PRINT "CORRECT:"; TAB( 20);R
550  PRINT "ERRORS:"; TAB( 20);W
560  REM  ASK FOR ANOTHER
570 Q2$ = "PRACTICE": GOSUB 4500
580  RETURN
```

Figure 7.3: Program Listing: Mathematics Practice I Subroutines

ADDITION

Description

The addition drill presents two random numbers at a time. You must enter the answer from left to right, which will sometimes mean that you must first work out the problem on a piece of paper. The program selects the largest number you work with from the answer you enter when the menu first appears.

Example

The sample display in Figure 7.4 shows an incorrect addition. Note that six problems have been done; four out of six answers have been correct. The Addition program is listed in Figure 7.5.

```
ADDITION DRILL


                    3
                +   5
                  ----
CORRECT ANSWER IS:  8

TYPE ANSWER:      ?8

YOU ARE RIGHT!
PROBLEMS DONE:     6
CORRECT:           4
ERRORS:            2



----------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN? (Y OR N)
```

Figure 7.4: Screen Display: Addition

```
590  REM ----------------
600 N$ = "ADDITION DRILL"
610  REM ----------------
620  GOSUB 7500
630 N1 =  INT ( RND (1) * X)
640 N2 =  INT ( RND (1) * X)
650 SN$ = "+"
660  REM  DISPLAY QUESTION AND
670  REM  GET ANSWER
680  GOSUB 320
690  REM  SET FLAG
700 S = 0
710  IF A = N1 + N2 THEN S = 1
720  PRINT  SPC( 4 -  LEN ( STR$ (N1 + N2)))N1 + N2
730  GOSUB 460
740  IF YN$ = "N" THEN  RETURN
750  GOTO 590
```

Figure 7.5: Program Listing: Addition

SUBTRACTION

Description

The subtraction drill is similar to the addition drill. The program's selection method for the two numbers guarantees that a positive answer always results. Remember, practice makes perfect—try lots of problems.

Example

The screen display in Figure 7.6 shows an incorrect subtraction. Note that two out of three problems have been correctly answered. Figure 7.7 displays the program listing for the subtraction drill.

```
SUBTRACTION DRILL


                  20
                -  12
                ----
CORRECT ANSWER IS:     8

TYPE ANSWER:       ?8

YOU ARE RIGHT!
PROBLEMS DONE:     3
CORRECT:           2
ERRORS:            1



----------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN? (Y OR N)
```

Figure 7.6: Screen Display: Subtraction

```
760  REM ------------------
770 N$ = "SUBTRACTION DRILL"
780  REM ------------------
790  GOSUB 7500
800 N1 =  INT ( RND (1) * X)
810 N2 =  INT ( RND (1) * X)
820 N3 = N1
830  IF N1 > N2 THEN 850
840 N1 = N2:N2 = N3
850 SN$ = "-"
860  REM  DISPLAY QUESTION AND
870  REM  GET ANSWER
880  GOSUB 320
890  REM  SET FLAG
900 S = 0
910  IF A = N1 - N2 THEN S = 1
920  PRINT  SPC( 4 -  LEN ( STR$ (N1 - N2)))N1 - N2
930  REM  FINISH UP
940  GOSUB 460
950  IF YN$ = "N" THEN  RETURN
960  GOTO 760
```

Figure 7.7: Program Listing: Subtraction

MULTIPLICATION

Description

This program allows you to practice multiplying two numbers. The program generates random numbers; you choose the largest number you want to work with at the time you select the multiplication drill from the menu. This program displays the answer on a single line. If you are multiplying by two or more digits, you may wish to use the Long Multiplication program, which appears later in this chapter. The Long Multiplication program displays the intermediate steps necessary to compute the answer.

Example

Figure 7.8 shows a screen display for the Multiplication program. This display shows that four out of five problems have been done correctly. Figure 7.9 is the program listing for the multiplication drill.

```
MULTIPLICATION DRILL


                    9
                X   3
                ----
CORRECT ANSWER IS:   27

TYPE ANSWER:       ?27

YOU ARE RIGHT!
PROBLEMS DONE:      5
CORRECT:            4
ERRORS:             1



----------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN? (Y OR N)
```

Figure 7.8: Screen Display: Multiplication

```
970  REM ----------------------
980 N$ = "MULTIPLICATION DRILL"
990  REM ----------------------
1000  GOSUB 7500
1010 N1 =  INT ( RND (1) * X)
1020 N2 =  INT ( RND (1) * X)
1030 SN$ = "X"
1040  REM  DISPLAY QUESTION AND
1050  REM  GET ANSWER
1060  GOSUB 320
1070  REM  SET FLAG
1080 S = 0
1090  IF A = N1 * N2 THEN S = 1
1100  PRINT  SPC( 4 -  LEN ( STR$ (N1 * N2)))N1 * N2
1110  REM  FINISH UP
1120  GOSUB 460
1130  IF YN$ = "N" THEN  RETURN
1140  GOTO 970
```

Figure 7.9: Program Listing: Multiplication

DIVISION

Description

The fourth mathematics drill allows you to practice dividing two whole numbers. Again, you choose the largest number you want to divide by (the denominator) when the Division program is selected from the Mathematics Practice I Menu. The program generates numbers which guarantee that a whole number results from your division. Note that the largest number you work with is the largest answer in this program.

Example

Figure 7.10 shows a screen display for the Division program. Note that this is the fifth problem completed, and that all the answers have been correct. The listing for the Division program appears in Figure 7.11.

```
DIVISION DRILL


                  126
                /  18
                ----
CORRECT ANSWER IS:    7

TYPE ANSWER:       ?7

YOU ARE RIGHT!
PROBLEMS DONE:      5
CORRECT:            5
ERRORS:             0



----------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN? (Y OR N)
```

Figure 7.10: Screen Display: Division

```
1150  REM ----------------
1160 N$ = "DIVISION DRILL"
1170  REM ----------------
1180  GOSUB 7500
1190 N1 =  INT ( RND (1) * X)
1200 N2 =  INT ( RND (1) * X)
1210  IF N2 = 0 THEN 1200
1220 N1 = N1 * N2
1230 SN$ = "/"
1240  REM  DISPLAY QUESTION AND
1250  REM  GET ANSWER
1260  GOSUB 320
1270  REM  SET FLAG
1280 S = 0
1290  IF A = N1 / N2 THEN S = 1
1300  PRINT  SPC( 4 -  LEN ( STR$ (N1 / N2)))N1 / N2
1310  REM  FINISH UP
1320  GOSUB 460
1330  IF YN$ = "N" THEN  RETURN
1340  GOTO 1150
```

Figure 7.11: Program Listing: Division

MATHEMATICS PRACTICE II MENU

Mathematics Practice II consists of two advanced mathematics drills. The first program allows you to practice adding numbers in columns. The second program allows you to practice multiplying large numbers using long multiplication. For both programs, work out the answer with pencil and paper and then enter it into the computer. Note that the multiplication program provides all intermediate steps. Figure 7.12 is a screen display of the Mathematics Practice II menu. Figure 7.13 is a listing of the menu program.

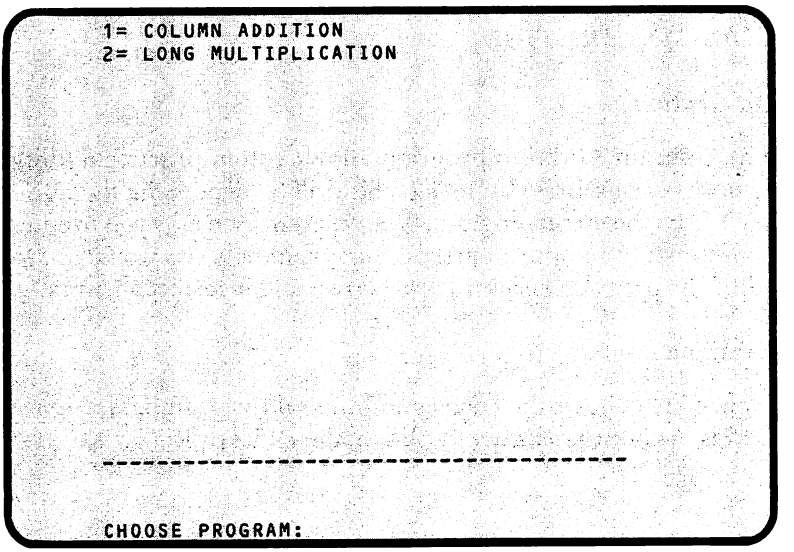


Figure 7.12: Screen Display: Mathematics Practice II Menu

```
100  REM ------------------------
110 N$ = "MATHEMATICS PRACTICE II"
120  REM ------------------------
130  GOSUB 7500: REM INITIALIZE
140  FOR X = 1 TO 1000: NEXT
150 X$(1) = "COLUMN ADDITION"
160 X$(2) = "LONG MULTIPLICATION"
170  REM SET UP MENU, DISPLAY IT
180 N = 2: GOSUB 8500
190  ON X GOSUB 210,610
200  GOTO 100
```

Figure 7.13: Program Listing: Mathematics Practice II Menu

COLUMN ADDITION

Description

The Column Addition program allows you to practice adding columns of numbers. You select this drill by depressing the number 1 from the menu program. The program then asks you to enter both the largest number and how many numbers you would like to add. The program keeps track of correct and incorrect answers.

Example

Figure 7.14 shows the correct addition of five numbers. Figure 7.15 is the program listing for this mathematics drill.

```
210  REM -----------------
220 N$ = "COLUMN ADDITION"
230  REM -----------------
240  GOSUB 7500: REM INITIALIZE
250 Q1$ = ""
260 Q2$ = "WHAT IS THE LARGEST NUMBER"
270 Q3$ = "YOU WOULD LIKE TO ADD"
280  GOSUB 6000
290  INPUT LG
300 Q2$ = "HOW MANY NUMBERS"
310  GOSUB 6000
320  INPUT NU
```

Figure 7.15: Program Listing: Column Addition (continues)

```
COLUMN ADDITION

                      9
                      9
                     14
                     20
                     19
                    ------
CORRECT ANSWER:      71

YOUR ANSWER:       ?71

YOU ARE RIGHT!
PROBLEMS DONE:      1
CORRECT:            1
ERRORS:             0



----------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN? (Y OR N)
```

Figure 7.14: Screen Display: Column Addition

```
330 R = 0:W = 0
340  GOSUB 7500: REM INITIALIZE
350 T = 0
360  FOR I = 1 TO NU
370 N1 =  INT ( RND (1) * LG)
380 T = T + N1
390  PRINT  TAB( 20);
400  PRINT  SPC( 6 -  LEN ( STR$ (N1)));N1
410  NEXT I
420  PRINT  TAB( 20);"------"
430  VTAB NU + 6
440  PRINT "YOUR ANSWER:"; TAB( 19);
450  INPUT A
460  VTAB NU + 4
470  PRINT "CORRECT ANSWER:"; TAB( 20)
480  PRINT  SPC( 6 -  LEN ( STR$ (T)));T
490  VTAB NU + 8:S = 0
500  IF A = T THEN S = 1
510  PRINT "YOU ARE ";
520  IF S = 1 THEN R = R + 1: PRINT "RIGHT!"
530  IF S <  > 1 THEN W = W + 1: PRINT "WRONG."; CHR$ (7)
540  PRINT "PROBLEMS DONE:"; TAB( 20);R + W
550  PRINT "CORRECT:"; TAB( 20);R
560  PRINT "ERRORS:"; TAB( 20);W
570  REM  ASK FOR ANOTHER
580 Q2$ = "PRACTICE": GOSUB 4500
590  IF YN$ = "N" THEN  RETURN
600  GOTO 340
```

Figure 7.15: Program Listing: Column Addition

LONG MULTIPLICATION

Description

We now present a program that allows you to practice multiplying numbers using long multiplication. You select this program from the Mathematics Practice II Menu by pressing the 2 on your keyboard. You then enter the number of digits you wish to work with. The program presents you with two numbers to multiply; you work out your answer by hand and enter the final answer. The screen displays the correct answer and the steps that are necessary to compute the answer. By reviewing the various steps, you can see exactly where you make your mistakes.

Example

Figure 7.16 shows the result of multiplying two large numbers. Note that all of the intermediate results are displayed. Figure 7.17 is the Long Multiplication program listing.

```
610  REM ---------------------
620 N$ = "LONG MULTIPLICATION"
630  REM ---------------------
640  GOSUB 7500: REM INITIALIZE
650 Q1$ = ""
660 Q2$ = "HOW MANY DIGITS"
670 Q3$ = "WOULD YOU LIKE TO MULTIPLY? "
680  GOSUB 6000
690  INPUT N
700  GOSUB 7500
710 N1 = 0:N2 = 0
720  FOR I = 0 TO N - 1
730 N1(I) =  INT ( RND (1) * 10)
```

Figure 7.17: Program Listing: Long Multiplication (continues)

```
LONG MULTIPLICATION

                      8801
                      4447
                  ----------
                     61607
                    352040
                   3520400
                  35204000
                  ----------
                  39138047



TYPE YOUR ANSWER: ->?39138047

YOU ARE RIGHT!

----------------------------------------
WOULD YOU LIKE TO
MULTIPLY
AGAIN? (Y OR N)
```

Figure 7.16: Screen Display: Long Multiplication

```
740 N1 = N1 + N1(I) * 10 ^ I
750 N2(I) =  INT ( RND (1) * 10)
760 N2 = N2 + N2(I) * 10 ^ I
770  NEXT I
780 N1 =  INT (N1):N2 =  INT (N2)
790  HTAB 20: PRINT  SPC( 10 -  LEN ( STR$ (N1)));N1
800  HTAB 20: PRINT  SPC( 10 -  LEN ( STR$ (N2)));N2
810  PRINT  TAB( 20);"----------"
820  VTAB 16
830  PRINT "TYPE YOUR ANSWER: ->";
840  INPUT A
850  VTAB 6
860  FOR I = 0 TO N - 1
870 S1 = 10 -  LEN ( STR$ (N1 * N2(I) * 10 ^ I))
880  HTAB 20: PRINT  SPC( S1);N1 * N2(I) * 10 ^ I
890  NEXT I
900  PRINT  TAB( 20);"----------"
910 S1 = 10 -  LEN ( STR$ (N1 * N2))
920  PRINT  TAB( 20); SPC( S1);N1 * N2: PRINT
930  VTAB 18: PRINT "YOU ARE ";
940 S = 0
950  IF A = N1 * N2 THEN S = 1
960  IF S = 1 THEN  PRINT "RIGHT!"
970  IF S <  > 1 THEN  PRINT "WRONG."; CHR$ (7)
980 Q2$ = "MULTIPLY": GOSUB 4500
990  IF YN$ = "N" THEN  RETURN
1000  GOTO 700
```

Figure 7.17: Program Listing: Long Multiplication

MATHEMATICS PRACTICE III MENU

The final set of mathematics drills allows you to practice adding, subtracting, multiplying, and dividing fractions. You work out your answers by hand and enter the final answer. All answers in this section are reduced to least-common-denominator form; conversions to mixed numbers are not made. Each program keeps track of your correct and incorrect answers.

The four fraction programs work from a common menu. Figure 7.18 displays the Mathematics Practice III Menu; Figure 7.19 is the listing of the menu program.

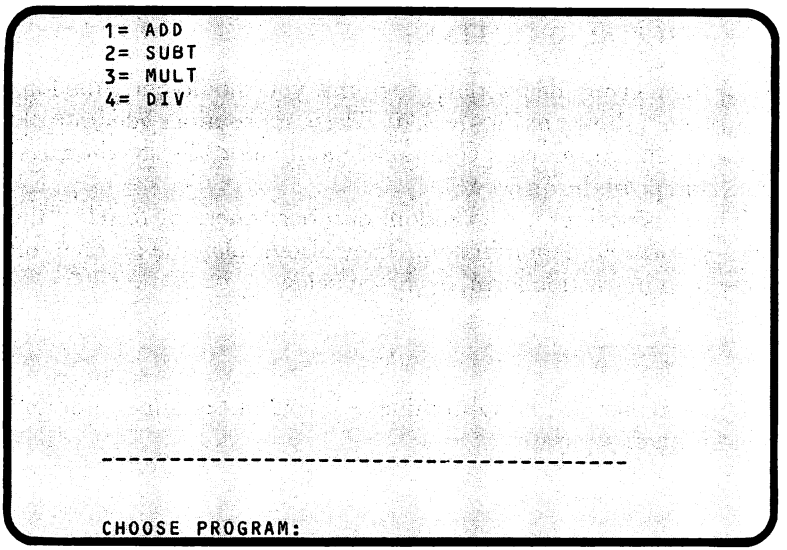


Figure 7.18: Screen Display: Mathematics Practice III Menu

```
100  REM -------------------------
110 N$ = "MATHEMATICS PRACTICE III"
120  REM -------------------------
130  GOSUB 7500: REM INITIALIZE
140  FOR X = 1 TO 1000: NEXT
150  REM  USE MENU TO CHOOSE
160  REM  TYPE OF DRILL
170 X$(1) = "ADD":X$(2) = "SUBT"
180 X$(3) = "MULT":X$(4) = "DIV"
190 N = 4: GOSUB 8500
200 Y = X
210 R = 0:W = 0
220 Q1$ = "WHAT IS THE LARGEST DENOMINATOR"
230 Q2$ = "YOU WOULD LIKE TO"
240 Q3$ = X$(Y)
250  GOSUB 6000
260  PRINT  TAB( 20);: INPUT X
270  ON Y GOSUB 920,1130,1360,1580
280  GOTO 100
```

Figure 7.19: Program Listing: Mathematics Practice III Menu

FRACTION SUBROUTINES

Prior to entering the fraction programs, you should enter the four subroutines shown in Figure 7.20. These subroutines generate the fractions and the greatest common denominator, display the question, and print the answer.

The subroutines use a random number generator to select the numerators and denominators. They also display the problem on the screen in fraction form, as shown below:

$$\frac{4}{5} + \frac{5}{6}$$

When you answer, the subroutines ask you for the numerator and denominator separately.

```
290  REM ----------------------
300  REM THIS ROUTINE GENERATES
310  REM THE NUMERATORS AND
320  REM DENOMINATORS
330  REM ----------------------
340  DEF  FN ND(X) =  INT ( RND (1) * X)
350 N1 =  FN ND(X): IF N1 = 0 THEN 350
360 N2 =  FN ND(X): IF N2 = 0 THEN 360
370 D1 =  FN ND(X): IF D1 = 0 THEN 370
380 D2 =  FN ND(X): IF D2 = 0 THEN 380
390  REM BE SURE SUBT IS POSITIVE
```

Figure 7.20: Program Listing: Fraction Subroutines (continues)

```
400  IF N1 / D1 >  = N2 / D2 THEN 440
410  REM OTHERWISE SWAP
420 N3 = N2:N2 = N1:N1 = N3
430 D3 = D2:D2 = D1:D1 = D3
440  RETURN
450  REM -----------------
460  REM  ROUTINE TO FIND
470  REM  GREATEST COMMON
480  REM  DENOMINATOR
490  REM -----------------
500 U = N3:V = D3
510 G = U - V *  INT (U / V)
520  IF G = 0 THEN  RETURN
530 U = V
540 V = G
550  GOTO 510
560  REM ----------------------
570  REM  DISPLAY QUESTION AND
580  REM  GET ANSWER
590  REM ----------------------
600  VTAB 4: HTAB 20
610 S1 = 3 -  LEN ( STR$ (N1)):S2 = 3 -  LEN ( STR$ (N2))
620  PRINT  SPC( S1);N1; TAB( 26); SPC( S2);N2
630  PRINT  TAB( 20);"--- ";SN$;" ---"
640 S1 = 3 -  LEN ( STR$ (D1)):S2 = 3 -  LEN ( STR$ (D2))
650  PRINT  TAB( 20); SPC( S1);D1; TAB( 26); SPC( S2);D2
660  VTAB 13
670  PRINT "TYPE NUMERATOR:   ";
680  INPUT AN
690  PRINT "TYPE DENOMINATOR: ";
700  INPUT AD: PRINT
710  VTAB 8: PRINT "CORRECT ANSWER IS:";
720  RETURN
730  REM ----------------------
740  REM   FINISH THE PROBLEM
750  REM ----------------------
760 S1 = 3 -  LEN ( STR$ (N3)):S2 = 3 -  LEN ( STR$ (D3))
770  PRINT  TAB( 20); SPC( S1);N3
780  PRINT  TAB( 20);"---"
790  PRINT  TAB( 20); SPC( S2);D3
800 S = 0
810  IF AN = N3 AND AD = D3 THEN S = 1
820  VTAB 16: PRINT "YOU ARE ";
830  REM  UPDATE SCORE
840  IF S = 1 THEN  PRINT "RIGHT!":R = R + 1
850  IF S <  > 1 THEN  PRINT "WRONG."; CHR$ (7):W = W + 1
860  PRINT "PROBLEMS DONE:"; TAB( 20);R + W
870  PRINT "CORRECT:"; TAB( 20);R
880  PRINT "ERRORS:"; TAB( 20);W
890  REM  ASK FOR ANOTHER
900 Q2$ = "PRACTICE": GOSUB 4500
910  RETURN
```

Figure 7.20: Program Listing: Fraction Subroutines

FRACTION ADDITION

Description

The Fraction Addition drill allows you to practice adding two fractions. You select this program from the fraction program menu by striking the 1 on your keyboard. You must also enter the largest denominator you want to work with. (Don't be afraid of large denominators!)

Example

Figure 7.21 shows the addition of two fractions. Note that this is the second problem of a series. Figure 7.22 is the program listing for the addition drill.

```
ADDITION DRILL


                    2     3
                   --- + ---
                    4     6

CORRECT ANSWER IS:  1
                   ---
                    1


TYPE NUMERATOR:   ?1
TYPE DENOMINATOR: ?1

YOU ARE RIGHT!
PROBLEMS DONE:     2
CORRECT:           1
ERRORS:            1
---------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN? (Y OR N)
```

Figure 7.21: Screen Display: Fraction Addition

```
920  REM ----------------
930 N$ = "ADDITION DRILL"
940  REM ----------------
950  GOSUB 7500: REM INITIALIZE
960  REM  GEN. NUMS AND DENOMS
970  GOSUB 290
980 SN$ = "+"
990  REM  DISPLAY QUESTION
1000  REM  AND GET ANSWER
1010  GOSUB 560
1020  REM  BEFORE CLEANUP
1030 N3 = D2 * N1 + D1 * N2
1040 D3 = D1 * D2
1050  REM  FIND GREATEST
1060  REM  COMMON DENOMINATOR
1070  GOSUB 450
1080 N3 = N3 / V:D3 = D3 / V
1090  REM  FINISH UP
1100  GOSUB 730
1110  IF YN$ = "N" THEN  RETURN
1120  GOTO 920
```

Figure 7.22: Program Listing: Fraction Addition

FRACTION SUBTRACTION

Description

You can practice subtracting fractions with this program. You select the program by pressing 2 as you run the fraction drill menu. You also select the largest denominator you want to work with. Like the subtraction drill in the first menu, this program generates its numbers so that a positive answer always results.

Example

Figure 7.23 shows a screen display for the Fraction Subtraction program. Figure 7.24 is the program listing.

```
SUBTRACTION DRILL

                    9     5
                   --- - ---
                    7     9

CORRECT ANSWER IS:  46
                   ---
                    63


TYPE NUMERATOR:   ?46
TYPE DENOMINATOR: ?63

YOU ARE RIGHT!
PROBLEMS DONE:     1
CORRECT:           1
ERRORS:            0
----------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN? (Y OR N)
```

Figure 7.23: Screen Display: Fraction Subtraction

```
1130  REM ------------------
1140 N$ = "SUBTRACTION DRILL"
1150  REM ------------------
1160  GOSUB 7500: REM INITIALIZE
1170  REM  GEN. NUMS AND DENOMS
1180  GOSUB 290
1190  REM  BEFORE CLEANUP
1200 N3 = D2 * N1 - D1 * N2
1210  REM  DISALLOW ZERO
1220  IF N3 = 0 THEN 1180
1230 SN$ = "-"
1240  REM  DISPLAY QUESTION AND
1250  REM  GET ANSWER
1260  GOSUB 560
1270 D3 = D1 * D2
1280  REM  FIND GREATEST
1290  REM  COMMON DENOMINATOR
1300  GOSUB 450
1310 N3 = N3 / V:D3 = D3 / V
1320  REM  FINISH UP
1330  GOSUB 730
1340  IF YN$ = "N" THEN  RETURN
1350  GOTO 1130
```

Figure 7.24: Program Listing: Fraction Subtraction

FRACTION MULTIPLICATION

Description

The Fraction Multiplication program allows you to practice multiplying two fractions. You select this program from the fraction menu by pressing the number 3. Fraction multiplication is very easy to learn, and it comes in handy—don't be afraid to try this drill.

Example

Figure 7.25 shows a screen display for the Fraction Multiplication program. Note that two out of four problems have been correctly answered. Figure 7.26 is the Fraction Multiplication program listing.

```
MULTIPLICATION DRILL


                      5     2
                     --- X ---
                      6     6

CORRECT ANSWER IS:    5
                     ---
                     18


TYPE NUMERATOR:   ?5
TYPE DENOMINATOR: ?18

YOU ARE RIGHT!
PROBLEMS DONE:     4
CORRECT:           2
ERRORS:            2
----------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN? (Y OR N)
```

Figure 7.25: Screen Display: Fraction Multiplication

```
1360  REM ----------------------
1370 N$ = "MULTIPLICATION DRILL"
1380  REM ----------------------
1390  GOSUB 7500: REM INITIALIZE
1400  REM  GEN. NUMS AND DENOMS
1410  GOSUB 290
1420 SN$ = "X"
1430  REM  DISPLAY QUESTION AND
1440  REM  GET ANSWER
1450  GOSUB 560
1460  REM  BEFORE CLEANUP
1470 N3 = N1 * N2
1480 D3 = D1 * D2
1490  REM  FIND GREATEST
1500  REM  COMMON DENOMINATOR
1510  GOSUB 450
1520 N3 = N3 / V
1530 D3 = D3 / V
1540  REM  FINISH UP
1550  GOSUB 730
1560  IF YN$ = "N" THEN  RETURN
1570  GOTO 1360
```

Figure 7.26: Program Listing: Fraction Multiplication

FRACTION DIVISION

Description

The final fraction program allows you to practice dividing fractions. This program is chosen by pressing the 4 on your keyboard from the menu program. Dividing fractions is the same as multiplying fractions, except that you invert the second fraction (the denominator). Practice with this program will sharpen your skills and make fraction division a less formidable task for you.

Example

Figure 7.27 shows a screen display for the Fraction Division program. Figure 7.28 is the program listing.

```
DIVISION DRILL


                      6     3
                     --- / ---
                      4     5

CORRECT ANSWER IS:    5
                     ---
                      2


TYPE NUMERATOR:   ?5
TYPE DENOMINATOR: ?2

YOU ARE RIGHT!
PROBLEMS DONE:     1
CORRECT:           1
ERRORS:            0
---------------------------------------
WOULD YOU LIKE TO
PRACTICE
AGAIN? (Y OR N)
```

Figure 7.27: Screen Display: Fraction Division

```
1580  REM ----------------
1590 N$ = "DIVISION DRILL"
1600  REM ----------------
1610  GOSUB 7500: REM INITIALIZE
1620  REM  GEN. NUMS AND DENOMS
1630  GOSUB 290
1640  REM  BEFORE CLEANUP
1650 N3 = N1 * D2
1660 D3 = D1 * N2
1670  REM  FIND GREATEST
1680  REM  COMMON DENOMINATOR
1690  GOSUB 450
1700 N3 = N3 / V
1710 D3 = D3 / V
1720 SN$ = "/"
1730  REM  DISPLAY QUESTION AND
1740  REM  GET ANSWER
1750  GOSUB 560
1760  REM  FINISH UP
1770  GOSUB 730
1780  IF YN$ = "N" THEN  RETURN
1790  GOTO 1580
```

Figure 7.28: Program Listing: Fraction Division

CENTRAL SUBROUTINES

These subroutines allow us to create useful programs that are very short and thus easy to enter. Interested users should read Appendix B to see how these subroutines can be used in other applications.

This appendix contains the program listings for the central subroutines used throughout the book. The programs contain many remark statements to help you understand the functions of each subroutine.

You can save retyping these subroutines many times by saving them together in one disk file. Add them to your application programs as explained in Chapter 1. Remember that you need type only the first REM in each subroutine.

```
4000  REM          "ADREC"
4010  REM ----------------------
4020  REM    ADD RECORDS TO FILE
4030  REM
4040  REM    THIS ROUTINE ADDS
4050  REM    RECORDS TO A FILE
4060  REM
4070  REM   CALLING PARAMETERS:
4080  REM     NF=NUMBER OF FIELDS
4090  REM         IN A REC (MAX=8)
4100  REM     F$(L)=NAMES OF FIELDS
4110  REM     F$=NAME OF FILE TO ADD TO
4120  REM
4130  REM   RETURNED PARAMETERS:
4140  REM     NONE
4150  REM
4160  REM   ADDS RECORDS TILL
4170  REM   TERMINATED BY USER
4180  REM   8 FIELDS ARE
4190  REM   CREATED EVEN IF BLANK
4200  REM
4210  REM ----------------------
4220 N1 = 0:D$ =  CHR$ (4)
4230 K = 1: GOSUB 7500
4240 N1 = N1 + 1:Q1$ = "ENTER DATA"
4250 Q2$ = ""
4260  FOR L = 1 TO NF:Q3$ = F$(L): GOSUB 7000: NEXT L
4270  FOR L = 1 TO 8:S$(N1,L) = P$(L): NEXT L
4280  IF N1 = 10 THEN  GOSUB 4340:N1 = 0
4290 Q1$ = "ADD ANOTHER RECORD?"
4300 Q2$ = "":Q3$ = "YES OR NO (Y/N)"
4310  GOSUB 6000: GOSUB 9500: IF YN$ = "Y" THEN 4230
4320  IF N1 > 0 THEN  GOSUB 4340
4330  RETURN
4340  GOSUB 5000
4350  PRINT D$;"OPEN";FB$: PRINT D$;"OPEN";F$
4370  PRINT D$;"READ";FB$: INPUT N2: PRINT D$
4380  PRINT D$;"WRITE";F$: PRINT N1 + N2: PRINT D$
4390  FOR L = 1 TO N2: PRINT D$;"READ";FB$
4400  INPUT P1$,P2$,P3$,P4$,P5$,P6$,P7$,P8$
4410  PRINT D$: PRINT D$;"WRITE";F$: PRINT P1$
4420  PRINT P2$: PRINT P3$: PRINT P4$: PRINT P5$
4430  PRINT P6$: PRINT P7$: PRINT P8$: NEXT L
4440  PRINT D$;"WRITE";F$: FOR L = 1 TO N1
4450  FOR L1 = 1 TO 8: PRINT S$(L,L1): NEXT L1,L
4460  PRINT D$;"CLOSE": RETURN
4488  PRINT D$;"DELETE";F$: PRINT D$;"OPEN";F$
```

Figure A.1: Add Record Subroutine

```
4500  REM          "ANOTH"
4510  REM ----------------------
4520  REM   ASK FOR ANOTHER RUN
4530  REM
4540  REM  THIS ROUTINE ASKS THE USER WHETHER TO CONT.
4560  REM
4570  REM  CALLING PARAMETERS:
4580  REM    Q2$=STRING TO SPEC. WHAT OPERATION TO DO
4600  REM
4610  REM  RETURNED PARAMETERS:
4620  REM    YN$=A "Y" OR "N" FOR YES OR NO
4640  REM
4650  REM ----------------------
4660 Q1$ = "WOULD YOU LIKE TO"
4670 Q3$ = "AGAIN? (Y OR N)"
4680  REM  PRESENT DIALOG
4690  GOSUB 6000
4700  REM  ASK YES OR NO
4710  GOSUB 9500
4720  RETURN
```

Figure A.2: Request to Run Again Subroutine

```
5000  REM         "BAKFIL"
5010  REM ----------------------
5020  REM      BACK UP A FILE
5040  REM  THIS ROUTINE CREATES A BACKUP COPY OF A FILE
5070  REM  CALLING PARAMETERS:
5080  REM    F$=NAME OF FILE TO BACKUP
5090  REM    FB$=NAME OF BACKUP FILE
5110  REM ----------------------
5120  PRINT  CHR$ (4);"OPEN";F$
5130  PRINT  CHR$ (4);"OPEN";FB$
5140  PRINT  CHR$ (4);"DELETE";FB$
5150  PRINT  CHR$ (4);"OPEN";FB$
5160  PRINT  CHR$ (4);"READ";F$
5170  INPUT N: PRINT  CHR$ (4)
5180  PRINT  CHR$ (4);"WRITE";FB$
5190  PRINT N: PRINT  CHR$ (4)
5200  FOR R = 1 TO N
5210  PRINT  CHR$ (4);"READ";F$
5220  INPUT P1$,P2$,P3$,P4$,P5$,P6$,P7$,P8$
5230  PRINT  CHR$ (4)
5240  PRINT  CHR$ (4);"WRITE";FB$
5250  PRINT P1$: PRINT P2$: PRINT P3$: PRINT P4$
5260  PRINT P5$: PRINT P6$: PRINT P7$: PRINT P8$
5270  PRINT  CHR$ (4)
5280  NEXT R
5290  PRINT  CHR$ (4);"CLOSE"
5300  RETURN
```

Figure A.3: Create Backup File Subroutine

```
5500  REM         "DELREC"
5510  REM ----------------------
5520  REM      DELETE RECORD
5530  REM
5540  REM   THIS ROUTINE DELETES
5550  REM   A RECORD FROM A FILE
5560  REM
5570  REM   CALLING PARAMETERS:
5580  REM     FB$=NAME OF BACKUP
5590  REM      FILE TO CREATE
5600  REM     F$=NAME OF FILE TO
5610  REM      DELETE FROM
5620  REM
5630  REM   RETURNED PARAMETERS:
5640  REM     L=-1 IF RECORD NOT
5650  REM      FOUND
5660  REM
5670  REM   USER IS ASKED
5680  REM   FOR RECORD NO.
5690  REM   BACKUP FILE IS
5700  REM   CREATED
5710  REM
5720  REM ----------------------
5730 N$ = "DELETE RECORD": GOSUB 7500:D$ =  CHR$ (4)
5740 Q1$ = "NUMBER OF RECORD TO DELETE"
5750 Q2$ = "":Q3$ = "": GOSUB 6000
5760  INPUT L:L = L + 1: REM GET REC. NO.
5770  GOSUB 5000: REM BACKUP FILE
5780  PRINT D$;"OPEN";FB$: PRINT D$;"OPEN";F$
5790  PRINT D$;"READ";FB$
5800  INPUT N: IF N < L THEN L =  - 1: GOTO 5990
5810  PRINT D$: PRINT D$;"WRITE";F$: PRINT N - 1
5820  PRINT D$: IF L = 1 THEN 5900
5830  FOR I = 1 TO L - 1
5840  PRINT D$;"READ";FB$
5850  INPUT P1$,P2$,P3$,P4$,P5$,P6$,P7$,P8$
5860  PRINT D$: PRINT D$;"WRITE";F$
5870  PRINT P1$: PRINT P2$: PRINT P3$: PRINT P4$
5880  PRINT P5$: PRINT P6$: PRINT P7$: PRINT P8$
5890  PRINT D$: NEXT I
5900  PRINT D$;"READ";FB$
5910  INPUT P1$,P2$,P3$,P4$,P5$,P6$,P7$,P8$
5920  IF N = L THEN 5990
5930  FOR I = L + 1 TO N: PRINT D$;"READ";FB$
5940  INPUT P1$,P2$,P3$,P4$,P5$,P6$,P7$,P8$
5950  PRINT D$: PRINT D$;"WRITE";F$
5960  PRINT P1$: PRINT P2$: PRINT P3$: PRINT P4$
5970  PRINT P5$: PRINT P6$: PRINT P7$: PRINT P8$
5980  PRINT D$: NEXT I
5990  PRINT D$;"CLOSE": RETURN
```

Figure A.4: Delete Record Subroutine

```
6000  REM          "DIALOG"
6010  REM ----------------------
6020  REM          DIALOGUE
6030  REM
6040  REM   CALLING PARAMETERS:
6050  REM     Q1$,Q2$,Q3$ ARE
6060  REM     QUESTIONS TO DISPLAY
6070  REM
6080  REM   RETURNED PARAMETERS:
6090  REM     NONE
6100  REM
6110  REM   UNUSED QUESTION STRINGS
6120  REM   SHOULD BE SET TO NULL ( "" )
6130  REM
6140  REM ----------------------
6150  REM   CLEAR DIALOG AREA
6160  REM   DRAW BORDER, BLANK ROWS
6170  FOR I = 20 TO 23
6180  REM PICK ROW
6190  VTAB I: HTAB 1
6200  FOR J = 1 TO 39
6210  IF I = 20 THEN  PRINT "-";
6220  IF I <  > 20 THEN  PRINT " ";
6230  NEXT J
6240  NEXT I
6250  VTAB 21: HTAB 1
6260  PRINT Q1$: REM FIRST QUES
6270  PRINT Q2$: REM 2ND QUES
6280  PRINT Q3$;: REM 3RD QUES
6290  RETURN
```

Figure A.5: Display Dialog Subroutine

```
6500  REM          "INCH"
6510  REM ----------------------
6520  REM     INPUT CHARACTER
6530  REM
6540  REM   CALLING PARAMETERS:
6550  REM     NONE
6560  REM
6570  REM   RETURNED PARAMETERS:
6580  REM     CH$=INPUT CHARACTER
6590  REM
6600  REM ----------------------
6610  REM WAIT FOR INPUT CHAR
6620  GET CH$
6630  RETURN
```

Figure A.6: Input Character Subroutine

```
7000  REM          "INDAT"
7010  REM ----------------------
7020  REM        INPUT DATA
7030  REM
7040  REM   CALLING PARAMETERS:
7050  REM     Q1$,Q2$=USER INSTRUCTIONS
7070  REM     Q3$=NAME OF DATA ITEMS
7080  REM
7090  REM   RETURNED PARAMETERS:
7100  REM     P$(K)=ARRAY OF DATA ITEMS
7110  REM
7120  REM   ONE LINE AT A TIME IS INPUT
7140  REM   FIRST CALL WITH K=1
7150  REM   K WILL BE UPDATED AUTOMATICALLY
7160  REM
7170  REM ----------------------
7180  REM   ASK QUESTIONS
7190  GOSUB 6000
7200  REM   GET DATA
7210  INPUT P$(K)
7220  REM ECHO QUESTION & ANSWER
7230  VTAB K + 2: HTAB 3
7240  PRINT Q3$; TAB( 25);P$(K)
7250  REM   UPDATE INDEX
7260 K = K + 1
7270  RETURN
```

Figure A.7: Input Data Subroutine

```
7500  REM          "INIT"
7510  REM ----------------------
7520  REM    INITIALIZE DISPLAY
7530  REM
7540  REM   THIS ROUTINE CLEARS
7550  REM   THE SCREEN AND PRINTS
7560  REM   THE PROGRAM TITLE
7570  REM
7580  REM   CALLING PARAMETERS:
7590  REM     N$=TITLE OF PROGRAM
7600  REM
7610  REM   RETURNED PARAMETERS:
7620  REM     NONE
7630  REM
7640  REM ----------------------
7650  REM   CLEAR SCREEN
7660  HOME
7670  REM   TITLE
7680  PRINT N$: PRINT
7690  RETURN
```

Figure A.8: Initialize Program and Display Subroutine

```
8000  REM          "INPAR"
8010  REM ----------------------
8020  REM     INPUT PARAMETERS
8030  REM
8040  REM   CALLING PARAMETERS:
8050  REM     Q1$,Q2$ ARE USER INSTRUCTIONS
8060  REM     Q3$=NAME OF DATA ITEM
8070  REM
8080  REM   RETURNED PARAMETERS:
8090  REM     PAR(K)=DATA ITEM OBTAINED
8100  REM
8110  REM   CALL WITH K=1 FOR
8120  REM   FIRST DATA ITEM
8130  REM   K IS UPDATED AUTOMATICALLY
8140  REM
8150  REM ----------------------
8160  REM   ASK QUESTIONS
8170  GOSUB 6000
8180  REM   GET VALUE
8190  INPUT PAR(K)
8200  REM ECHO QUESTION & ANSWER
8210  VTAB K + 2: HTAB 3
8220  PRINT Q3$; TAB( 25);PAR(K)
8230 K = K + 1
8240  RETURN
```

Figure A.9: Input Program Parameters Subroutine

```
8500  REM          "MENU"
8510  REM ----------------------
8520  REM       MENU PROGRAM
8530  REM
8540  REM  THIS PROGRAM DISPLAYS
8550  REM  A MENU AND CHOOSES A PROGRAM
8560  REM
8570  REM  CALLING PARAMETERS:
8580  REM    N=NO. OF MENU ITEMS
8590  REM    X$(I)=ARRAY OF PROGRAM NAMES
8600  REM
8610  REM  RETURNED PARAMETERS:
8620  REM    X=PROGRAM NUMBER CHOSEN
8630  REM
8640  REM ----------------------
8650  HOME
8660  REM DISPLAY MENU
8670  FOR I = 1 TO N
8680  IF I < 10 THEN  PRINT I;"= ";X$(I)
8690  IF I = 10 THEN  PRINT "0= ";X$(I)
8700  NEXT I
8710  REM  ASK QUESTIONS
8720 Q1$ = "":Q2$ = "":Q3$ = "CHOOSE PROGRAM: "
8730  GOSUB 6000
8740  REM  INPUT CHARACTER
8750  GOSUB 6500
8760 X =  VAL (CH$): IF X = 0 THEN X = 10
8770  REM  SEE IF CHAR IN RANGE
8780  IF X >  = 1 AND X <  = N THEN  RETURN
8790 Q1$ = "ILLEGAL CHOICE, CHOOSE AGAIN"
8800  GOSUB 6000
8810  GOTO 8750
```

Figure A.10: Display Menu Subroutine

```
9000  REM          "PAUSE"
9010  REM ----------------------
9020  REM      WAIT FOR A KEY
9030  REM
9040  REM  THIS ROUTINE WAITS FOR
9050  REM  USER TO STRIKE A KEY
9060  REM
9070  REM  CALLING PARAMETERS:
9080  REM    NONE
9090  REM
9100  REM  RETURNED PARAMETERS:
9110  REM    NONE
9120  REM
9130  REM ----------------------
9140 Q1$ = "STRIKE ANY KEY TO CONTINUE"
9150 Q2$ = "":Q3$ = ""
9160  REM  CALL DIALOG PROGRAM
9170  GOSUB 6000
9180  REM  WAIT FOR KEY
9190  GET X$
9200  RETURN
```

Figure A.11: Pause Subroutine

```
9500  REM          "YESNO"
9510  REM ----------------------
9520  REM          YES-NO
9530  REM
9540  REM  CALLING PARAMETERS:
9550  REM    NONE
9560  REM
9570  REM  RETURNED PARAMETERS:
9580  REM    YN$="Y" OR "N"
9590  REM
9600  REM ----------------------
9610  REM WAIT FOR KEY TO BE STRUCK
9620  GET YN$
9630  IF YN$ = "Y" OR YN$ = "N" THEN 9660
9640  REM  NOT YES OR NO
9650  GOTO 9620
9660  PRINT YN$: RETURN
```

Figure A.12: Yes or No Subroutine

B

HOW TO USE THE CENTRAL SUBROUTINES

Readers who would like to improve their programming skills should try writing their own programs. You can save a lot of time and effort by using the central subroutines in this book as part of your programs. This appendix will help you learn to use these central subroutines. We review here several of the subroutines and how they function. We also include sample programs that show how the subroutines can be used.

MENU SET-UP

The menu set-up subroutine begins at line 8500. This subroutine displays a menu on the screen and waits for the user to press a number key. The menu program then returns a value, which you use to select the program you want to run.

Figures B.1 and B.2 show the program listing and screen display for a sample menu selection. We set up the array X$(I) with the names of the available programs. We then set N= to the number of program choices and call the menu subroutine. The subroutine displays the names of the programs and waits for the user to make a selection. The subroutine then returns the value of the selected program. The returned value is used in a GOSUB statement to select a program.

```
100  REM --------------------
110 N$ = "MENU DEMONSTRATION"
120  REM --------------------
130  GOSUB 7500: REM INITIALIZE
140  REM SET UP MENU ARRAY
150 X$(1) = "PROGRAM 1"
160 X$(2) = "PROGRAM 2"
170 X$(3) = "PROGRAM 3"
180 X$(4) = "PROGRAM 4"
190  REM DISPLAY MENU
200 N = 4: GOSUB 8500
210  VTAB 6: HTAB 1
220  REM SELECT PROGRAM
230  ON X GOTO 240,250,260,270
240  PRINT X$(1);" CHOSEN": END
250  PRINT X$(2);" CHOSEN": END
260  PRINT X$(3);" CHOSEN": END
270  PRINT X$(4);" CHOSEN": END
```

Figure B.1: Program Listing: Sample Menu Selection

```
1= PROGRAM 1
2= PROGRAM 2
3= PROGRAM 3
4= PROGRAM 4

----------------------------------------

CHOOSE PROGRAM:
```

Figure B.2: Screen Display: Sample Menu Selection

CHOOSING ANOTHER PROGRAM

All the programs in this book use the ANOTH subroutine. This subroutine asks the user whether he would like to run the program again. The subroutine displays three lines of data. The ANOTH subroutine generates the first and third data lines. You specify the second line when you call the subroutine.

The ANOTH routine returns a Y or an N in the variable YN$. The program tests this variable to decide whether it will run the program again.

Figures B.3 and B.4 show the program listing and a screen display for the ANOTH demonstration. Note that the sample program displays the question DEMONSTRATE THIS ROUTINE.

```
280  REM ----------------------------------------
290 N$ = "DEMONSTRATION OF 'ANOTHER' SUBROUTINE"
300  REM ----------------------------------------
310  GOSUB 7500: REM INITIALIZE
320 Q2$ = "DEMONSTRATE THIS ROUTINE"
330  GOSUB 4500: REM ASK FOR ANOTH
340  IF YN$ = "Y" THEN 280
350  END
```

Figure B.3: Program Listing: Demonstration of ANOTH Subroutine

```
DEMONSTRATION OF 'ANOTHER' SUBROUTINE

---------------------------------------
WOULD YOU LIKE TO
DEMONSTRATE THIS ROUTINE
AGAIN? (Y OR N)
```

Figure B.4: Screen Display: Demonstration of ANOTH Subroutine

THE DIALOG SUBROUTINE

We use the DIALOG subroutine to display three lines of text at the bottom of the screen. You have the flexibility to generate three different questions; three questions or instructions will handle most programs you design. We often give an instruction on the first line, with questions on the second and third lines. This routine also draws a border and clears leftover text from previous questions. The routine exits with the cursor after the third question. Figures B.5 and B.6 show a program listing and screen display of a DIALOG demonstration.

```
360  REM ---------------------------------------
370 N$ = "DEMONSTRATION OF 'DIALOG' SUBROUTINE"
380  REM ---------------------------------------
390  GOSUB 7500: REM INITIALIZE
400  REM SET UP QUESTIONS
410 Q1$ = "THIS IS FIRST QUESTION"
420 Q2$ = "": REM BLANK QUESTION
430 Q3$ = "THIS IS LAST QUESTION"
440  REM SEND OUT THE QUESTIONS
450  GOSUB 6000
460  END
```

Figure B.5: Program Listing: Demonstration of DIALOG Subroutine

```
DEMONSTRATION OF 'DIALOG' SUBROUTINE









---------------------------------------
THIS IS FIRST QUESTION

THIS IS LAST QUESTION
```

Figure B.6: Screen Display: Demonstration of DIALOG Subroutine

DATA INPUT SUBROUTINE

The INPAR program handles data input for many of the programs in this book and is easily used to create additional programs. You begin the data input process by clearing the screen with the INIT subroutine and setting the index K equal to 1. You are now ready to input program parameters. For each parameter, set Q3$ equal to the name of the parameter and call the subroutine at 8000. The INPAR subroutine asks the parameter name at the bottom of the screen and waits for input. The parameter then appears at the top of the screen. INPAR stores parameters in the array PAR(K) in the order of entry. Figures B.7 and B.8 show a program listing and screen display of the Data Input program.

```
470  REM --------------------------
480 N$ = "DATA INPUT DEMONSTRATION"
490  REM --------------------------
500  GOSUB 7500: REM INITIALIZE
510 K = 1
520  REM SET UP DIALOG AREA
530 Q1$ = "ENTER DATA":Q2$ = ""
540  REM GET FIRST ITEM
550 Q3$ = "RETAIL PRICE ($)": GOSUB 8000
560  REM GET SECOND ITEM
570 Q3$ = "DISCOUNT (%)": GOSUB 8000
580  PRINT : PRINT "PRICE ="; TAB( 25);
590 P = PAR(1) - PAR(2) / 100 * PAR(1)
600  REM ROUND OFF TO 2 PLACES
610 P =  INT (P * 100 + .5) / 100
620  PRINT P
630  END
```

Figure B.7: Program Listing: Data Input Demonstration

```
DATA INPUT DEMONSTRATION

  RETAIL PRICE ($)      300
  DISCOUNT (%)          10

PRICE =                 270

----------------------------------------
ENTER DATA

DISCOUNT (%)?10
```

Figure B.8: Screen Display: Data Input Demonstration

USEFUL PRINTER SUBROUTINES

You will enjoy these printer routines if you own an Epson printer, which can output data in a number of interesting type styles. You can print in normal, expanded, condensed, emphasized, double-strike, and emphasized and double-strike modes. Figure C.1 shows the different printing modes.

You can use the subroutines in this appendix to gain access to your printer's different type styles. Emphasized mode is very useful for letter printing; condensed mode prints 132-column reports on 80-column paper.

Figure C.2 shows the menu selection program for printer control. Figure C.3 lists subroutines you can use to select the different type styles. Figure C.4 lists the program that uses the printer subroutines to produce the sample type styles in Figure C.1.

```
      PRINTING MODES AVAILABLE ON EPSON PRINTER


THIS IS NORMAL PRINTING.

THIS IS CONDENSED PRINTING.

T H I S   I S   E X P A N D E D   P R I N T I N G .

THIS IS EMPHASIZED PRINTING.

THIS IS DOUBLE-STRIKE PRINTING.

THIS IS EMPHASIZED AND DOUBLE-STRIKE PRINTING.
```

Figure C.1: Printing Modes Available on the Epson Printer

```
100  REM ------------------------
110 N$ = "PRINTER CONTROL FUNCTIONS"
120  REM ------------------------
130  GOSUB 7500
140  REM SET UP MENU ARRAY
150 X$(1) = "CONDENSED MODE ON"
160 X$(2) = "CONDENSED MODE OFF"
170 X$(3) = "EXPANDED MODE ON"
180 X$(4) = "EXPANDED MODE OFF"
190 X$(5) = "EMPHASIZED MODE ON"
200 X$(6) = "EMPHASIZED MODE OFF"
210 X$(7) = "DOUBLE-STRIKE ON"
220 X$(8) = "DOUBLE-STRIKE OFF"
230 X$(9) = "PRINT SAMPLE TEXT"
240  REM  DISPLAY MENU
250 N = 9: GOSUB 8500
260  ON X GOSUB 280,330,380,430,480,530,580,630,680
270  GOTO 100
```

Figure C.2: Program Listing: Menu Selection for Printer Control

```
280  REM --------------------
290 N$ = "CONDENSED PRINT ON"
300  REM --------------------
310  PRINT  CHR$ (15);
320  RETURN
330  REM ---------------------
340 N$ = "CONDENSED PRINT OFF"
350  REM ---------------------
360  PRINT  CHR$ (18);
370  RETURN
380  REM -------------------
390 N$ = "EXPANDED PRINT ON"
400  REM -------------------
410  PRINT  CHR$ (14);
420  RETURN
430  REM --------------------
440 N$ = "EXPANDED PRINT OFF"
450  REM --------------------
460  PRINT  CHR$ (20);
470  RETURN
480  REM ---------------------
490 N$ = "EMPHASIZED PRINT ON"
500  REM ---------------------
510  PRINT  CHR$ (27) +  CHR$ (69);
520  RETURN
530  REM ----------------------
540 N$ = "EMPHASIZED PRINT OFF"
550  REM ----------------------
560  PRINT  CHR$ (27) +  CHR$ (70);
570  RETURN
580  REM ------------------
590 N$ = "DOUBLE-STRIKE ON"
600  REM ------------------
610  PRINT  CHR$ (27) +  CHR$ (71);
620  RETURN
630  REM -------------------
640 N$ = "DOUBLE-STRIKE OFF"
650  REM --------------------
660  PRINT  CHR$ (27) +  CHR$ (72);
670  RETURN
```

Figure C.3: Program Listing: Subroutines to Select Type Styles

```
680  REM -------------
690 N$ = "SAMPLE TEXT"
700  REM -------------
710  HOME
720  PRINT "TURN PRINTER ON,"
730  PRINT "STRIKE ANY KEY WHEN READY"
740  GOSUB 9000: REM WAIT FOR CHAR
750  REM USE EMPHASIZED PRINT IN TITLE
760  PR# 1
770  GOSUB 480
780  PRINT
790  PRINT "     PRINTING MODES AVAILABLE ON EPSON PRINTER"
800  GOSUB 530
810  PRINT : PRINT : PRINT
820 T1$ = "THIS IS ":T2$ = "PRINTING."
830  PRINT T1$;"NORMAL ";T2$: PRINT
840  GOSUB 280
850  PRINT T1$;"CONDENSED ";T2$: PRINT
860  GOSUB 330: GOSUB 380
870  PRINT T1$;"EXPANDED ";T2$: PRINT
880  GOSUB 430: GOSUB 480
890  PRINT T1$;"EMPHASIZED ";T2$: PRINT
900  GOSUB 530: GOSUB 580
910  PRINT T1$;"DOUBLE-STRIKE ";T2$: PRINT
920  GOSUB 630: GOSUB 480: GOSUB 580
930  PRINT T1$;"EMPHASIZED AND DOUBLE-STRIKE ";T2$
940  GOSUB 530: GOSUB 630: PR# 0
950  END
```

Figure C.4: Program Listing: Subroutines Used to Select Type Styles Shown in Figure C.1

The SYBEX Library

MORE BOOKS FOR YOUR APPLE

THE EASY GUIDE TO YOUR APPLE II®

by Joseph Kascmer 160 pp., illustr., Ref. 0-122

A friendly introduction to using the Apple II, II plus, and the new IIe.

BASIC EXERCISES FOR THE APPLE®

by J. P. Lamoitier 250 pp., 90 illustr., Ref. 0-084

Teaches Apple BASIC through actual practice, using graduated exercises drawn from everyday applications.

APPLE II® BASIC HANDBOOK

by Douglas Hergert, 144 pp., Ref. 0-115

A complete listing with descriptions and instructive examples of each of the Apple II BASIC keywords and functions. A handy reference guide, organized like a dictionary.

YOUR FIRST APPLE II® PROGRAM

by Rodnay Zaks 150 pp. illustr., Ref. 0-136

A fully illustrated, easy-to-use introduction to APPLE BASIC programming. Will have the reader programming in a matter of hours.

MASTERING VISICALC®

by Douglas Hergert 217 pp., 140 illustr., Ref. 0-090

Explains how to use the VisiCalc "electronic spreadsheet" functions and provides examples of each. Makes using this powerful program simple.

DOING BUSINESS WITH VISICALC®

by Stanley R. Trost 260 pp., Ref. 0-086

Presents accounting and management planning applications—from financial statements to master budgets; from pricing models to investment strategies.

VISICALC® FOR SCIENCE AND ENGINEERING

by Stanley R. Trost & Charles Pomernacki 225 pp., illustr., Ref. 0-096

More than 50 programs for solving technical problems in the science and engineering fields. Applications range from math and statistics to electrical and electronic engineering.

BASIC FOR BUSINESS

by Douglas Hergert 224 pp., 15 illustr., Ref. 0-080

A logically organized, no-nonsense introduction to BASIC programming for business applications. Includes many fully-explained accounting programs, and shows you how to write them.

EXECUTIVE PLANNING WITH BASIC

by X. T. Bui 196 pp., 19 illustr., Ref. 0-083

An important collection of business management decision models in BASIC, including Inventory Management (EOQ), Critical Path Analysis and PERT, Financial Ratio Analysis, Portfolio Management, and much more.

BASIC PROGRAMS FOR SCIENTISTS AND ENGINEERS

by Alan R. Miller 318 pp., 120 illustr., Ref. 0-073

This book from the "Programs for Scientists and Engineers" series provides a library of problem-solving programs while developing proficiency in BASIC.

CELESTIAL BASIC

by Eric Burgess 300 pp., 65 illustr., Ref. 0-087

A collection of BASIC programs that rapidly complete the chores of typical astronomical computations. It's like having a planetarium in your own home! Displays apparent movement of stars, planets and meteor showers.

PROGRAMMING THE 6502

by Rodnay Zaks 386 pp., 160 illustr., Ref. 0-046

Assembly language programming for the 6502, from basic concepts to advanced data structures.

INTRODUCTION TO COMPUTERS

DON'T (or How to Care for Your Computer)

by Rodnay Zaks 214 pp., 100 illustr., Ref. 0-065

The correct way to handle and care for all elements of a computer system, including what to do when something doesn't work.

YOUR FIRST COMPUTER

by Rodnay Zaks 258 pp., 150 illustr., Ref. 0-045

The most popular introduction to small computers and their peripherals: what they do and how to buy one.

INTERNATIONAL MICROCOMPUTER DICTIONARY

120 pp., Ref. 0-067

All the definitions and acronyms of microcomputer jargon defined in a handy pocket-size edition. Includes translations of the most popular terms into ten languages.

FROM CHIPS TO SYSTEMS: AN INTRODUCTION TO MICROPROCESSORS

by Rodnay Zaks 552 pp., 400 illustr., Ref. 0-063

A simple and comprehensive introduction to microprocessors from both a hardware and software standpoint: what they are, how they operate, how to assemble them into a complete system.

INTRODUCTION TO WORD PROCESSING

by Hal Glatzer 205 pp., 140 illustr., Ref. 0-076

Explains in plain language what a word processor can do, how it improves productivity, how to use a word processor and how to buy one wisely.

INTRODUCTION TO PASCAL (Including UCSD Pascal™)

by Rodnay Zaks 420 pp., 130 illustr., Ref. 0-066

A step-by-step introduction for anyone wanting to learn the Pascal language. Describes UCSD and Standard Pascals. No technical background is assumed.

THE PASCAL HANDBOOK

by Jacques Tiberghien 486 pp., 270 illustr., Ref. 0-053

A dictionary of the Pascal language, defining every reserved word, operator, procedure and function found in all major versions of Pascal.

APPLE® PASCAL GAMES

by Douglas Hergert and Joseph T. Kalash 372 pp., 40 illustr., Ref. 0-074

A collection of the most popular computer games in Pascal, challenging the reader not only to play but to investigate how games are implemented on the computer.

INTRODUCTION TO THE UCSD p-SYSTEM™

by Charles W. Grant and Jon Butah 300 pp., 10 illustr., Ref. 0-061

A simple, clear introduction to the UCSD Pascal Operating System; for beginners through experienced programmers.

PASCAL PROGRAMS FOR SCIENTISTS AND ENGINEERS

by Alan R. Miller 374 pp., 120 illustr., Ref. 0-058

A comprehensive collection of frequently used algorithms for scientific and technical applications, programmed in Pascal. Includes such programs as curve-fitting, integrals and statistical techniques.

DOING BUSINESS WITH PASCAL

by Richard Hergert & Douglas Hergert 371 pp., illustr., Ref. 0-091

Practical tips for using Pascal in business programming. Includes design considerations, language extensions, and applications examples.

For a complete catalog of our publications:

2344 Sixth Street
Berkeley, California 94710
Tel: (415) 848-8233
Telex: 336311

APPLE II
BASIC PROGRAMS IN MINUTES

You don't have to be a programmer to program your Apple II, II plus, or IIe!

These versatile, ready-to-enter programs will enable you to perform over 65 home and business tasks with your Apple Computer. Use these handy programs for:

- Home Finances—
  - Future Value of an IRA
  - Building a College Fund
- Business Calculations—
  - Break-Even Point
  - Depreciation Schedules
- Real Estate—
  - Affordable Home Price
  - Effect of Accelerating Mortgage Payments
- Data Analysis—
  - Input and Plot Data
  - Compute Moving Averages
- Record Keeping—
  - Telephone & Mailing Lists
  - Automobile Log
- Education—
  - Arithmetic Drills

You will be surprised at how easy it is to enter and use these programs! Any one of them can be entered and ready to run in less than 10 minutes. No knowledge of BASIC programming is necessary to use this collection of useful programs.

ABOUT THE AUTHOR:

Stanley R. Trost is a principal in the firm SR Trost & Associates, Walnut Creek, CA. He consults on the use of computer systems for business, medical, and engineering applications. He earned a Master's degree in Electronic Engineering and an MBA. Mr. Trost is the author of several books, including *Doing Business with VisiCalc, Doing Business with SuperCalc, VisiCalc for Science and Engineering,* and *Useful BASIC Programs for the IBM PC.*

ISBN 0-89588-121-7